5-1-2002

Automatic indoor/outdoor scene classification

Navid Serrano

AUTOMATIC INDOOR/OUTDOOR SCENE CLASSIFICATION

by

Navid Serrano

A Thesis Submitted

in

Partial Fulfillment

of the

Requirements for the Degree of

MASTER OF SCIENCE

in

Electrical Engineering

Approved by: _____

Prof. Andreas E. Savakis, Thesis Advisor

_____

Prof. Sohail A. Dianat, Thesis Committee

_____

Prof. Raghuveer M. Rao, Thesis Committee

_____

Dr. Jiebo Luo, Thesis Committee

_____

Prof. Robert J. Bowman, Department Head

DEPARTMENT OF ELECTRICAL ENGINEERING
COLLEGE OF ENGINEERING
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK
MAY, 2002

AUTOMATIC INDOOR/OUTDOOR SCENE CLASSIFICATION

Navid Serrano, May 13, 2002

Rochester Institute of Technology

Thesis Abstract

AUTOMATIC INDOOR/OUTDOOR SCENE CLASSIFICATION

By Navid Serrano

Prof. Andreas E. Savakis, Chairperson of the Supervisory Committee
Department of Electrical Engineering

The advent and wide acceptance of digital imaging technology has motivated an upsurge in research focused on managing the ever-growing number of digital images. Current research in image manipulation represents a general shift in the field of computer vision from traditional image analysis based on low-level features (e.g. color and texture) to semantic scene understanding based on high-level features (e.g. grass and sky). One particular area of investigation is scene categorization, where the organization of a large number of images is treated as a classification problem. Generally, the classification involves mapping a set of traditional low-level features to semantically meaningful categories, such as indoor and outdoor scenes, using a classifier engine. Successful indoor/outdoor scene categorization is beneficial to a number of image manipulation applications, as indoor and outdoor scenes represent among the most general scene types. In content-based image retrieval, for example, a query for a scene containing a sunset can be restricted to images in the database pre-categorized as outdoor scenes. Also, in image enhancement, categorization of a scene as indoor vs. outdoor can lead to improved color balancing and tone reproduction.

Prior research in scene classification has shown that high-level information can, in fact, be inferred from low-level image features. Classification rates of roughly 90% have been reported using low-level features to predict indoor scenes vs. outdoor scenes. However, the high classification rates are often achieved by using computationally expensive, high-dimensional feature sets, thus limiting the practical implementation of such systems. To address this problem, a low complexity, low-dimensional feature set was extracted in a variety of configurations in the work presented here. Due to their excellent generalization performance, Support Vector Machines (SVMs) were used to manage the tradeoff between reduced dimensionality and increased classification accuracy. It was determined that features extracted from image subblocks, as opposed to the full image, can yield better classification rates when combined in a second stage. In particular, applying SVMs in two stages led to an indoor/outdoor classification accuracy of 90.2% on a large database of consumer photographs provided by Kodak. Finally, it was also shown that low-level and semantic features can be integrated efficiently using Bayesian networks for increased accuracy. Specifically, the integration of grass and sky semantic features with color and texture low-level features increased the indoor/outdoor classification rate to 92.8% on the same database of images.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 The Scene Classification Problem

The number of digital images to be processed, transmitted, and archived is increasing rapidly as the quality and variety of digital capture technology grow. Consequently, the need for efficient digital image management technology has also risen in importance. A concerted effort to develop technology capable of interpreting the content of digital imagery has also gained momentum. Successful interpretation of scene content—i.e. image understanding—has been and remains a very challenging problem in computer vision. It can be said that modern research in image understanding evolved from prior research in image analysis [1]. However, image understanding deals primarily with *high-level* (semantic) scene information including people, buildings, and natural scenery, for example, while image analysis has traditionally focused on *low-level* image content such as color and texture. Thus, the challenge in image understanding has been to move from low-level scene description to more meaningful semantic interpretation.

An important area of research in image understanding is scene classification. The goal in scene classification is to organize images categorically. Scene classification has been investigated primarily for image database management applications. Yet, any area involving image manipulation, such as digital photofinishing, stands to benefit from successful scene classification. Knowledge of the scene can help improve the performance of such digital photofinishing operations as color balance and tone reproduction, for instance.

The inference of semantic scene content from low-level features has traditionally been accomplished through statistical learning. Although this approach has been shown to be successful, it does have limitations. The limitations rest, in particular, on the reliance on classifier engines to infer the semantic content from low-level features. High classification accuracy can be achieved if there is consistency between the images in the training set. This, however, implies that the classification will be less accurate for images outside the training set—especially those with different scene characteristics. On the other hand, when large and diverse image sets are used to train the classifier engines, the classification accuracy also decreases considerably because there is insufficient discrimination with low-level features alone. Although

this is a pervading challenge in the pattern recognition field in general, there are encouraging signs. Specifically, the incorporation of high-level scene information together with the traditional low-level image features seems a promising approach to scene classification.

1.2 Background

1.2.1 *Scene Classification Research*

Research in scene classification has been largely driven by applications involving the manipulation of a large volume of imagery [2]. Many image database management systems have been proposed over the years including QBIC [3], Photobook [4], VisualSEEk [5], NeTra [6], and MARS [7]. Most of these systems are example-based systems in that the low-level features of an example query image are used to find images in the database with similar features. These low-level features generally include some form of color and texture image analysis. However, it is well known that images with entirely different semantic scene content can possess similar low-level features. For example, it is entirely possible for a scene containing a school bus to be confused with a sunset scene using color features, such as a color histogram. Thus, scene interpretation using low-level features alone is usually ineffective. It is for this reason that higher level features have been investigated as a means of improving content-based image retrieval and image understanding, in general.

The need for automatic semantic scene interpretation and subsequent indexing in image databases were noted in [4]. In [8], a paradigm for texture annotation was proposed, where a texture model derived from an image (or image region) is linked to a semantic descriptor, e.g. water. The inclusion of, for instance, shape and sketch, as well as their location and size was proposed in [3-7]. Yet, image primitives including color, texture, and shape, to name a few, alone cannot fully bridge the gap to semantic scene understanding. In [6], text annotations were used to describe scene content. Although text annotations can describe semantic scene content, the content can be varied, complex, and thus, subject to interpretation. Furthermore, the semantic description is not automatic; it depends on human action. These limitations, as well as the growing number of applications involving image manipulation, have motivated an upsurge in scene classification

research, where images are organized according to semantically meaningful categories. Scene classification may be supervised or unsupervised. In the unsupervised case, clustering techniques are used to group images according to their semantic scene content [9,10]. In the supervised case, scenes are classified using pre-defined semantic categories, e.g. indoor scenes vs. outdoor scenes.

1.2.2 *Indoor/Outdoor Scene Classification*

Because, in general, scene classification is challenging, the problem can be simplified by considering broad scene categories, such as indoor scenes vs. outdoor scenes. Clearly, other semantic categories could be explored and, in fact, categorization according to human preference has been investigated [11]. However, indoor/outdoor classification has received significant attention partially because of its importance in image enhancement, as discussed earlier. Szummer and Picard first explored indoor/outdoor scene classification and showed that high-level image content could be deduced through statistical classification of low-level image attributes [12]. Later research addressing indoor/outdoor classification includes the work of Paek et al [13], Vailaya et al [11,14], Savakis and Luo [15], and Guerin-Dugue and Oliva [16], to name a few. Although these approaches vary in many respects, some general trends can be discerned.

1.2.3 *Feature Extraction*

Almost all documented indoor/outdoor scene classification work involves some degree of low-level feature extraction. In general, indoor/outdoor classification approaches employ color, texture, and to a lesser extent, spatial frequency low-level features. These features are typically computed on either the entire image or image subblocks. The most common color features used for indoor/outdoor classification include color histograms [12,13,15] and color moments [11,14]. In terms of texture features, the Multiresolution Simultaneous Auto-Regressive (MSAR) model was explored in [11,12,14,15], edge direction histograms were considered in [11,13,14], and Local Dominant Orientation (LDO) distributions were used in [16]. Spectral features, such as discrete cosine transform (DCT) coefficients, have also been investigated, though they have been shown to be less useful [12]. In the case of [15,16], the features are extracted from the entire

image, whereas the other approaches in [11-14] extract the features from image subblocks. Finally, in [13], textual features are derived from text annotations and combined with image-based features for indoor/outdoor classification.

1.2.4 *Feature Classification*

It can be said that there are two dominating paradigms for the classification of low-level indoor/outdoor image features: 1) feature extraction and classification from image subblocks, and 2) feature extraction and classification from the full image. If the first approach is used, the subblock features can be concatenated in order to obtain a feature vector representing the full image, as in [11,14,17]. Classification of the concatenated feature vector then yields a full image indoor/outdoor label. Alternatively, the subblock features can be classified independently [12]. This implies that the resulting indoor/outdoor classification for each subblock must somehow be combined to produce an indoor/outdoor label corresponding to the full image. The second, less common, approach, as in [15,16], involves low-level feature extraction and subsequent feature extraction from the whole image.

In terms of specific classifier engines, the k-nearest neighbor (k-NN) classifier has been frequently used and shown to produce favorable results for indoor/outdoor classification [12,15,16]. The main drawback, however, is that k-NN classifiers are generally slow. A more efficient Bayesian classification approach was used in [11,14] to classify concatenated subblock features. However, concatenated feature vectors can be very high dimensional. Not surprisingly, the concatenated feature vectors used in [11,14] had on the order of 600 dimensions. High feature dimensionality can be problematic and is typically avoided.

1.2.5 *Feature Integration*

The indoor/outdoor classification methods described in [11,14,16] construct a feature vector corresponding to the entire image. The final indoor/outdoor assignment is produced by statistical classification of this feature vector. Thus, these methods rely on the classifier engine to map the

relationship between the low-level features and the semantic scene content. Approaches of this type have typically achieved indoor/outdoor classification rates slightly lower than 90%. Specifically, in [11,14], indoor/outdoor classification rates of 88.2% and 88.7% were reported for two independent image test sets. The work described in [16] included two other classes (*open* and *closed* scenes) in addition to the indoor and outdoor classes resulting in an overall recognition rate of 88.7%.

Many of the other referenced approaches involve additional feature integration (or interpretation) before making a final indoor/outdoor category assignment. In [12], two distinct classifiers were trained for color and texture features extracted from a 4 x 4 image tessellation. Consequently, two indoor/outdoor labels are assigned to each image subblock—one resulting from the color classifier and another from the texture classifier. The final indoor/outdoor classification was decided by majority vote. A final indoor/outdoor classification rate of 90.3% was reported on a database of 1324 images provided by Kodak. However, in [13] this method was evaluated on a distinct image database of 1300 news images and the indoor/outdoor classification rate dropped to 74.7%—significantly lower than 90.3%. This drop could be due to the fact that Kodak's image database contains a large number of images with duplicate scene content that might inflate the classification rates. This issue is revisited in section 6.1.

Other approaches have integrated the low-level image features described in the previous section with higher level features. For instance, Paek et al [13] proposed a unique approach to integrate low-level color and texture features computed over image subblocks together with textual information about the content of the scene. Using this technique, an indoor/outdoor classification rate of 86% was reported on a database of approximately 1300 news images.

In [15], Savakis and Luo computed color and texture features analogous to [12]. As in [12], a *k*-NN classifier was trained for both the color and texture features. However, the features were computed over the entire image, rather than image subblocks. Using this approach, indoor/outdoor classification rates of 74.2% and 82.2% were obtained for the color and texture features, respectively, on the Kodak image database (minus 15 ambiguous scenes) used in [12]. They also proposed a paradigm for the integration of the low-level color and texture features together with semantic scene information using Bayesian belief networks. In this approach, a

Bayes net was used to integrate the indoor/outdoor beliefs obtained using color and texture features together with semantic information that distinguishes outdoor scenes from indoor scenes, such as the presence of grass and/or sky regions in an image. This approach yielded a final indoor/outdoor classification rate of 90.1% using grass and sky ground truth and 84.7% using computed grass and sky features.

1.3 Thesis Contribution

The work presented in this thesis proposes an improved approach to indoor/outdoor scene classification. The key contributions of this thesis are the following:

1) Low dimensional, low complexity feature set
2) Application of Support Vector Machines (SVMs) to low-level feature classification
3) Application of SVMs to interpret subblock beliefs in a second classification stage
4) Increased overall indoor/outdoor classification by integrating low-level and semantic features using a Bayesian network

1.4 Thesis Outline

The thesis is organized as follows. Section 2 introduces the low-level feature set to be used for indoor/outdoor classification: color histograms, wavelet coefficients, and edge direction histograms. Section 3 discusses low-level feature classification using SVMs. In addition, a variety of configurations for feature extraction and classification are proposed. Section 4 describes the use of semantic feature detectors for indoor/outdoor classification. Section 5 presents an introduction to Bayesian belief networks and a paradigm for the integration of low-level and semantic features for indoor/outdoor classification. Section 6 delineates the results from each approach and compares the classification accuracy of the proposed system to existing work. Conclusions are drawn in section 7.

2. LOW-LEVEL FEATURE EXTRACTION

This section describes a series of low-level features for indoor/outdoor classification. Color and texture attributes constitute the most common low-level features in scene classification. Many color and texture analysis techniques exist, and a number of these have been proposed for indoor/outdoor classification. Popular color features include color moments [11,14,17], color histograms [12,15], color coherence vectors [11,14,17], and color correlograms [18]. Popular texture features include MSAR features [11,13,14,15,17], edge direction histograms [11,14,17], edge direction coherence vectors [11,14,17], and LDO distributions [16]. Although these features have been shown to be effective for indoor/outdoor classification, prior research has focused on successful resolution of the indoor/outdoor classification problem first, and tractability second. Hence, the low-level features considered here were assessed not only in terms of their effectiveness for indoor/outdoor classification but also in terms of their computational efficiency. Three different low-level features were evaluated: color histograms, wavelet texture features, and edge-direction histograms.

2.1 Color Features

It is well known that the spectral characteristics of natural and artificial illuminants can vary considerably [19]. Moreover, the primary light source in a scene will impact the color reproduction in a photograph, be it digital or analog. Thus, illuminant differences are among the most important scene characteristics that distinguish indoor scenes from outdoor scenes. Reconstruction of the spectral characteristics of the illumination source in a scene remains a difficult problem [20] and therefore, beyond the scope of scene classification. Instead, the goal in indoor/outdoor classification is to capture coarse scene illumination differences by using simple, well-established color features. Color histograms were used in the indoor/outdoor classification system proposed here because of their simplicity. Based on prior use of color histograms for indoor/outdoor classification [12,15], the precision of the histograms was reduced by a factor of 2 in order to improve efficiency. The choice of color space is of prime importance and is addressed in the following section.

2.1.1 *LST Color Space*

Many factors impact the selection of a color space. These factors may include the statistical distribution of color signals, the effect of noise in color coordinates, the effect of light source variations, and the effect of reflectance variations. Based on these considerations, the *LST* color space, also known as the *Ohta* color space or *T-space* [21], is a good choice. The *LST* color space has been shown to be useful in image processing applications [22] and is detailed below:

$$L = \frac{\alpha}{\sqrt{3}} \cdot (R + G + B) \tag{1}$$

$$S = \frac{\alpha}{\sqrt{2}} \cdot (R - B) \tag{2}$$

$$T = \frac{\alpha}{\sqrt{6}} \cdot (R - 2G + B) \tag{3}$$

For an 8-bit image, α is given by:

$$\alpha = \frac{255}{Max\{R, G, B\}} \tag{4}$$

The three *LST* components are orthogonal and have unit length. The *LST* color space is a luminance-chrominance representation and the color components are thus, approximately decorrelated. The *ST* (chrominance) components are intensity-invariant, meaning they do not vary with light source intensity changes. Furthermore, the *S* component of the *LST* color space represents daylight to tungsten illuminant variations. Finally, *LST* color channels represent the principal components of a large selection of natural *RGB* images.

2.1.2 *LST Color Histograms*

Having transformed an input *RGB* image to the *LST* color space, color histograms for each of the three channels were obtained. Let $h_L(b)$, $h_S(b)$, and $h_T(b)$ be the histograms corresponding to the

L, S, and T channels, respectively. Where, b=1,2,...,n_c represents the gray level bins, and n_c is the total number of bins per color histogram. The concatenated L, S, and T histograms constitute the color feature vector $\mathbf{x}_c$:

$$\mathbf{x}_c = [h_L(1), h_L(2),..., h_L(n_c), h_S(1), h_S(2),..., h_S(n_c), h_T(1), h_T(2),..., h_T(n_c)] \quad (5)$$

This color feature approach is analogous to that of [12,15] (including the choice of color space). The dimensionality of the color histogram feature vector is equal to $3n_c$. The number of bins per color histogram was set to n_c=16 for a feature dimensionality of 48. In [12,15], 32 bins per histogram were used for a feature dimensionality of 96. Therefore, the color features proposed here have half the dimensionality of the analogous features used in [12,15].

2.2 Wavelet Texture Features

Texture has long been an area of research in image analysis. The numerous texture features proposed over the years can be divided into five categories [23]: statistical, geometrical, structural, model-based, and signal processing features. Early approaches to texture analysis relied mostly on statistical features [24,25]. It has since been shown, however, that statistical methods do not adequately describe both local and global textural information [26]. Attention has thus concentrated on signal processing approaches, and in particular, multiresolution methods, which better preserve both local and global information [27,28]. Of particular note is the aforementioned MSAR model introduced by Mao and Jain [29]. As noted earlier, the MSAR model has been used considerably for indoor/outdoor classification [11,12,14,15,17]. Despite their popularity, MSAR texture features are computationally intensive. A more computationally efficient alternative is the wavelet transform [30].

The last decade, especially, has produced a host of work on the wavelet transform and its applications. Not originally envisaged as a tool for texture analysis *per se*, the applicability of the wavelet transform to texture analysis was first proposed in the pioneering work of Mallat [31]. Wavelet packets (or tree-structured wavelet transform) [32], and wavelet frames (or over-complete wavelets) [33] have also proven to be useful approaches to texture analysis. Moreover,

wavelet features performed favorably compared to other signal processing texture features, including MSAR features, in a recent evaluation [34]. Beyond texture analysis, the use of the wavelet transform has been extended to high-level scene analysis, as surveyed in [35]. Finally, it is worth noting that the wavelet transform has also been shown to exist in biological visual systems [36]. Due to their comparative computational efficiency and positive performance in texture analysis, wavelet features are considered here.

2.2.1 *Wavelet Transform*

Multiresolution analysis (MRA) refers to the process of decomposing a signal into a hierarchy of approximation and detail coefficients. Fundamental to MRA is the fact that the original signal can be reconstructed perfectly from the approximation and detail functions if the proper methodology is employed. Simple scaling alone is not feasible because frequency information is lost, thus making it impossible to reconstruct the original signal exactly. However, if the scaling is preceded by a filtering stage using a low-pass (LP) filter $h_0(n)$ and a high-pass (HP) filter $h_1(n)$, the original signal can be reconstructed perfectly by scaling the analyzed signal and then filtering it with a LP filter $g_0(n)$ and a HP filter $g_1(n)$. The decomposition can be done multiple times thus producing multiple scaled versions of the original signal—the aforementioned hierarchy.

The wavelet transform is a specialized form of MRA, where the signal is decomposed into a set of functions $\psi_{m,n}(t)$ defined by:

$$\psi_{m,n}(t) = 2^{m/2}\psi(2^m t - n) \tag{6}$$

The family of functions of Eq. (5) is generated by translating and dilating the mother wavelet $\psi(t)$. The mother wavelet satisfies $\int\psi(t)dt=0$, and is constructed from a scaling function $\phi(t)$ according to:

$$\phi(t) = \sum_{k=-\infty}^{\infty} h_0(k)\phi(2t - k) \tag{7}$$

$$\psi(t) = \sum_{k=-\infty}^{\infty} h_1(k)\phi(2t-k) \tag{8}$$

Where, $h_0(k)$ and $h_1(k)$ are the coefficients of the aforementioned LP and HP decomposition filters, respectively. If a discrete wavelet transform (DWT) is used, the filter coefficients themselves can be used instead of the continuous functions $\psi(t)$ and $\phi(t)$. Biorthogonal wavelet filter banks are a special class of DWT. In a biorthogonal wavelet representation, the decomposition filters $h_0(n)$ and $h_1(n)$ satisfy and are related by:

$$\sum_n h_0(n) = 1 \tag{9}$$

$$\sum_n (-1)^{n-1} h_1(n) = 2 \tag{10}$$

$$h_1(n) = (-1)^n h_0(1-n) \tag{11}$$

The reconstruction filters $g_0(n)$ and $g_1(n)$ are related to $h_0(n)$ and $h_1(n)$ according to:

$$g_0(n) = (-1)^n h_1(1-n) \tag{12}$$

$$g_1(n) = (-1)^{n-1} h_0(1-n) \tag{13}$$

The *biorthogonal* denomination stems from the fact that $h_0(n)$ and $g_1(n)$ are mutually orthogonal as are $h_1(n)$ and $g_0(n)$. In addition, biorthogonal filter banks are symmetric (i.e. have linear phase) and are approximately decorrelated.

The 2-D DWT is an extension of the 1-D case. In fact, a separable 2-D DWT can be easily implemented by applying the 1-D filters (described above) along the two dimensions of the signal. For a given image $f(x,y)$, a one-level, 2-D separable DWT decomposition can be implemented by convolving the rows and columns of $f(x,y)$ with combinations of $h_0(n)$ and $h_1(n)$, as shown below:

$$LL_1(x,y) = \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} f(i,j)h_0(i-2x)h_0(j-2y) \tag{14}$$

$$LH_1(x,y) = \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} f(i,j)h_0(i-2x)h_1(j-2y) \tag{15}$$

$$HL_1(x,y) = \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} f(i,j)h_1(i-2x)h_0(j-2y) \tag{16}$$

$$HH_1(x,y) = \sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} f(i,j)h_1(i-2x)h_1(j-2y) \tag{17}$$

Where, *LL*, *LH*, *HL* and *HH* stand for *low-low*, *low-high*, *high-low,* and *high-high*, respectively. Specifically, $LL_1(x,y)$ represent the approximation coefficients and $LH_1(x,y)$, $HL_1(x,y)$, and $HH_1(x,y)$ represent the detail coefficients. The subscript denotes the decomposition level (in this case 1). Figure 1 shows the 2-D separable DWT implementation graphically.

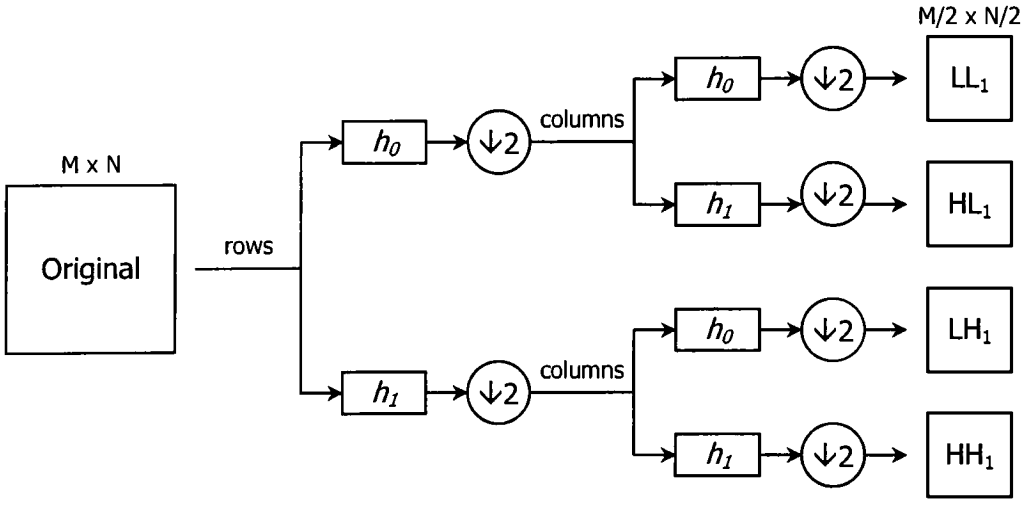


Figure 1. 2-D separable DWT implementation.

Wavelet coefficients are typically depicted using the pyramid structure shown in Figure 2. As can be seen from Figure 2, for each level of decomposition, it is the low-frequency coefficients that are further decomposed. This implies that the final decomposition at level K will consist of one coarse approximation (the final low-frequency coefficients) and K sets of detail (high frequency) coefficients.

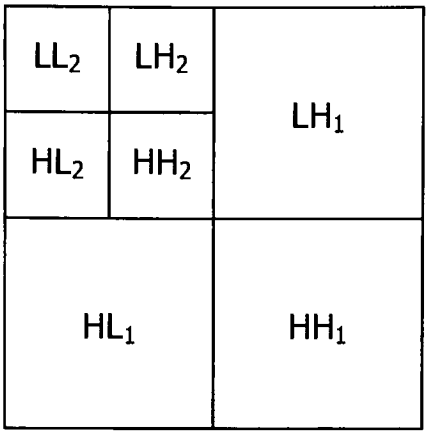


Figure 2. Two-level pyramid DWT structure.

2.2.2 *Wavelet Basis Selection*

An important point of consideration when employing the DWT is the selection of a wavelet basis (i.e. filter bank). Many wavelet filter banks exist, each possessing distinct properties. Biorthogonal wavelets (discussed in the previous section) have been shown to be useful for image compression [37] and will be used in the new JPEG2000 compression standard [38]. In the area of texture analysis, however, no formal evaluation of wavelet filter banks had been conducted until recently. Daubechies' filter banks have traditionally been a popular choice for wavelet texture analysis [32,39,40,41]. Yet, the selection of a wavelet basis for texture analysis has largely been subjective or arbitrary. In [42], Mojsilovic et al, attempt to define an optimal wavelet basis for texture characterization. Their study showed, first, that the selection of decomposition filters notably impacts texture characterization. Second, in comparing 19 orthogonal and biorthogonal wavelet filters, they found that biorthogonal filters outperformed orthogonal filters. A simple study described below provides further motivation for the use of biorthogonal, as opposed to, orthogonal filters.

A simple test image with edges oriented in four directions (0, 45, 90, and 135 degrees) is shown in Figure 3. The edges in the image have a width of one pixel. The simple test involves a wavelet decomposition of the test image using Daubechies' popular 4-tap orthonormal filter (*db4*) and the 5/3 biorthogonal filter (*bior5/3*). The LH_1 and HL_1 wavelet coefficients corresponding to the *db4* decomposition are shown in Figure 4a, and the LH_1 and HL_1 wavelet coefficients

corresponding to the *bior5/3* decomposition are shown in Figure 4b. As can be seen from Figure 4a, an interesting artifact appears when using the *db4* filter. Namely, the *db4* filter is unable to properly extract the edges oriented at 135 degrees (the first quadrant in the test image), whereas the *bior5/3* filters does. This is related to the frequency response of the *db4* filter and is clearly, an undesirable result.

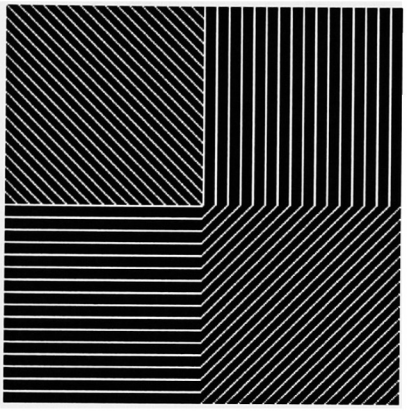
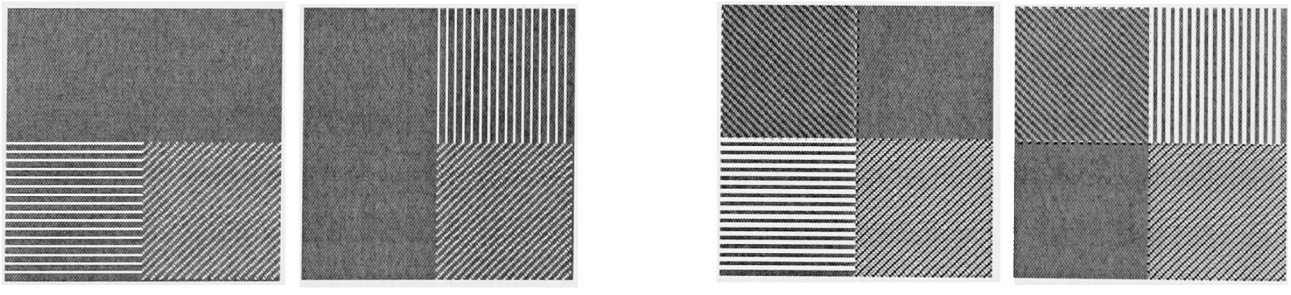


Figure 3. Edge pattern used for wavelet filter evaluation.



(a) (b)

Figure 4. LH_1 and HL_1 coefficients using the *db4* filter (a) and the *bior5/3* filter (b).

Another consideration is computational efficiency. Shorter filter lengths are, obviously, more computationally efficient as they require fewer numerical operations per pixel. Considering both computational tractability and optimal texture characterization, the *bior5/3* wavelet filters were selected for the proposed indoor/outdoor classification system. Finally, the *bior5/3* filter is one of two wavelet filter sets selected for inclusion in the JPEG2000 standard. Therefore, selecting this

filter may provide further efficiency gains when the indoor/outdoor classification system is implemented in conjunction with the JPEG2000 codec. The *bior5/3* decomposition filters $h_0(n)$ and $h_1(n)$ are shown in Eq. (18) and (19). In this case they are not shown in normalized form in order to illustrate the fact that they are integer coefficients.

$$h_0(n) = [-1 \quad 2 \quad 6 \quad 2 \quad -1] \tag{18}$$

$$h_1(n) = [1 \quad -2 \quad 1] \tag{19}$$

2.2.3 *Wavelet Texture Feature Extraction*

Let c_2, c_3, c_4, c_5, c_6, c_7, and c_8 represent the subband coefficients of the two-level wavelet decomposition as shown in Figure 5.

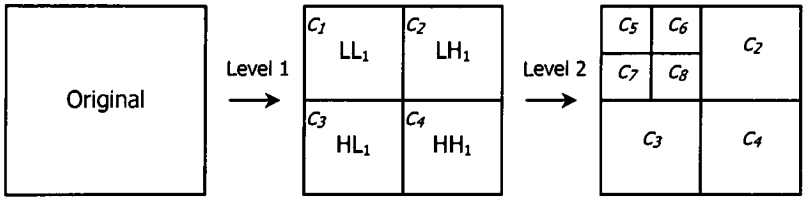


Figure 5. Coefficient labels for wavelet texture feature extraction.

As can be seen from Figure 5, c_2=$LH_1(x,y)$, c_3=$HL_1(x,y)$, c_4=$HH_1(x,y)$, c_5=$LL_2(x,y)$, c_6=$LH_2(x,y)$, c_7=$HL_2(x,y)$, and c_8=$HH_2(x,y)$. Coefficient set c_5 is a low-frequency approximation of the original signal. The other coefficients provide directionally correlated measures of the high-frequency signal content. Because natural textures contain mostly mid to high frequency information, the low-frequency coefficients are not inherently useful for texture description. Thus, rather than using the raw $LL_2(x,y)$ wavelet coefficients, set c_5 is redefined by filtering $LL_2(x,y)$ with the Laplacian filter:

$$c_5 = \sum_{i=1}^{M} \sum_{j=1}^{N} LL_2(i,j) h_L(x-i, y-j) \tag{20}$$

Where, M and N are the dimensions of $LL_2(x,y)$ and $h_L(x,y)$ is the Laplacian filter:

$$h_L(x,y) = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{21}$$

Because the Laplacian filter provides an isotropic measure of the high-frequency signal information, the filtered coefficient set c_5 can be regarded as a measure of non-directional high frequency energy in the image. Ultimately, the texture features are obtained by computing the subband energy for all wavelet coefficients (including the Laplacian filtered c_5 coefficients) according to the following general expression:

$$e_k = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} |c_k(i,j)|^2 \ , \ k = 2,3,\ldots 4K \tag{22}$$

Where, M and N are the image dimensions of coefficient c_k, and K is the number of decomposition levels (in this case 2). Therefore, seven wavelet texture features are obtained. This represents a reduction by a factor of 2 compared to the 15 MSAR features used in [11,12,14,15,17]. The wavelet texture feature vector $\mathbf{x}_t$ is defined as:

$$\mathbf{x}_t = [e_2, e_3, \ldots, e_8] \tag{23}$$

2.3 Edge Direction Features

It has been observed that many types of scenes have directional edge signatures. For instance, scenes containing man-made structures (e.g. city scenes) tend to have edges with dominant orientations. On the other hand, images with a predominance of natural scenery (e.g. landscape scenes) tend to have more randomly oriented edge content. Hence, it is reasonable to assume that directional edge features might be strong discriminators between certain types of scenes. Not surprisingly, Vailaya et al showed that edge direction histograms were, in fact, good discriminators of city vs. landscape scenes [17]. Edge direction histograms were also evaluated

for indoor/outdoor classification in [9,12,13]. Other directionally sensitive features have also been used for indoor/outdoor classification, including the LDO distributions introduced in [16]. In keeping with the goal of computational efficiency, edge direction histograms are considered here.

The process involves four basic steps: 1) edge detection, 2) computation of the edge magnitude and direction, and 3) selection of dominant edges, and 4) construction of the edge direction histogram. The method of Lee and Cok [43], which provides a framework for detecting boundaries in color images and estimating their magnitude and direction, is adopted here and summarized below.

2.3.1 *Edge Detection*

Given an image $f(x,y)$ with three color attributes (R,G,B), the individual color planes can be defined as:

$$r(x,y) = f(x,y) \in R \tag{24}$$

$$g(x,y) = f(x,y) \in G \tag{25}$$

$$b(x,y) = f(x,y) \in B \tag{26}$$

The horizontal edges can be obtained by convolving each of the above color planes with a derivative filter $h_x(x,y)$, thus yielding the partial derivatives:

$$\frac{\partial r}{\partial x} = \sum_{i=1}^{M}\sum_{j=1}^{N} r(i,j)h_x(x-i,y-j) \tag{27}$$

$$\frac{\partial g}{\partial x} = \sum_{i=1}^{M}\sum_{j=1}^{N} g(i,j)h_x(x-i,y-j) \tag{28}$$

$$\frac{\partial b}{\partial x} = \sum_{i=1}^{M}\sum_{j=1}^{N} b(i,j)h_x(x-i,y-j) \tag{29}$$

Where, M and N denote the row and column dimensions of $f(x,y)$. Similarly, the vertical edges are obtained by convolving the color planes with a derivative filter $h_y(x,y)$:

$$\frac{\partial r}{\partial y} = \sum_{i=1}^{M}\sum_{j=1}^{N} r(i,j)h_y(x-i, y-j) \tag{30}$$

$$\frac{\partial g}{\partial y} = \sum_{i=1}^{M}\sum_{j=1}^{N} g(i,j)h_y(x-i, y-j) \tag{31}$$

$$\frac{\partial b}{\partial y} = \sum_{i=1}^{M}\sum_{j=1}^{N} b(i,j)h_y(x-i, y-j) \tag{32}$$

The filters $h_x(x,y)$ and $h_y(x,y)$ can be any derivative filters. The well-known Prewitt filters were used here, where:

$$h_x(x,y) = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{33}$$

$$h_y(x,y) = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{34}$$

Finally, it should be noted that the image $f(x,y)$ was first smoothed using a Gaussian filter before applying filters (33) and (34) in order to suppress the effect of spurious edges.

2.3.2 *Edge Magnitude and Direction*

Having estimated the horizontal and vertical edges, a matrix **D** composed of the partial derivative values is defined as follows:

$$\mathbf{D} = \begin{bmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} \\ \frac{\partial g}{\partial x} & \frac{\partial g}{\partial y} \\ \frac{\partial b}{\partial x} & \frac{\partial b}{\partial y} \end{bmatrix} \tag{35}$$

The edge magnitude and direction can be obtained from principal component analysis of the matrix product $\mathbf{D}^T\mathbf{D}$, where the largest eigenvalue corresponds to the edge magnitude. The largest eigenvalue λ and thus the edge magnitude is given by:

$$\lambda = \frac{1}{2}\left[a + b + \sqrt{(a+b)^2 - 4(ab - c^2)}\right] \tag{36}$$

Where,

$$a = \left(\frac{\partial r}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial b}{\partial x}\right)^2 \tag{37}$$

$$b = \left(\frac{\partial r}{\partial y}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2 + \left(\frac{\partial b}{\partial y}\right)^2 \tag{38}$$

$$c = \frac{\partial r}{\partial x}\frac{\partial r}{\partial y} + \frac{\partial g}{\partial x}\frac{\partial g}{\partial y} + \frac{\partial b}{\partial x}\frac{\partial b}{\partial y} \tag{39}$$

The eigenvector corresponding to the largest eigenvalue λ provides the edge direction, which in turn can be used to compute the edge angle. The partial derivatives $\partial r/\partial x$, $\partial r/\partial x$, $\partial b/\partial x$, $\partial r/\partial y$, $\partial g/\partial y$, and $\partial b/\partial y$ have values associated with all points in $f(x,y)$. Thus, each pixel location in the original image will have a corresponding edge magnitude and direction.

2.3.3 *Dominant Edge Selection*

It is not meaningful to construct an edge direction histogram without first analyzing the edge magnitude and determining whether or not it is a dominant (i.e. significant) edge. Canny's popular edge detector [44] can be used for this purpose. Candidate points are those that are the local edge magnitude maxima along the corresponding edge direction. A candidate point is regarded as an edge if its edge magnitude is greater than a low threshold T_1 and is connected to at least one point that has an edge magnitude greater than a high threshold T_2. The thresholds T_1 and T_2 are typically determined empirically depending on the desired degree of edge selectivity.

2.3.4 *Edge Direction Histograms*

An edge direction histogram is accumulated only for those points that were marked as dominant edges according to the criteria of section 2.3.3. Let $h_e(b)$ be the edge direction histogram, where, b=1,2,...,n_e represents the edge direction bin element, and n_e is the total number of bins per edge direction histogram. In other words, n_e defines the number of edge angles to include in the histogram h_e. The edge direction feature vector $\mathbf{x}_e$ is given by:

$$\mathbf{x}_e = [h_e(1), h_e(2),\ldots, h_e(n_e), MN - \sum_b h_e(b)] \tag{40}$$

The last element in the feature vector (which is not part of the edge direction histogram) represents the number of non-edge points, where M and N are the row and column dimensions of the image or image subblock. The dimensionality of the edge direction histogram is $n_e + 1$. The number of bins per edge direction histogram was set to n_e=36, for a feature dimensionality of 37. This is roughly half the dimensionality of the analogous edge direction histograms of [11,14,17], where 5° angle intervals were used, yielding edge direction histograms with 72 bins (i.e. n_e=72).

3. LOW-LEVEL FEATURE CLASSIFICATION

3.1 Support Vector Machines

The Support Vector Machine (SVM) [45,46] is a new method of parameterization of functions, and therefore has application outside the realm of predictive learning. It has been called a *universal learning procedure* because it can be used to learn various representations such as neural networks, radial basis functions (RBF), polynomial estimators, etc. In the pattern recognition context, SVMs have been used for handwriting recognition [47], text categorization [48], and face detection [49]. SVMs have been shown to have equivalent or significantly better error rates than comparative classification methods [45]. One characteristic, in particular, separates SVMs from other classification paradigms—optimization of the separating hyperplane. This optimization (discussed in the following section) results in better generalization beyond the training data set. For these reasons, SVMs were used in the indoor/outdoor classification scheme proposed here.

3.1.1 *SVM Training*

In preceding sections, a color feature vector $\mathbf{x}_c$, a wavelet texture feature $\mathbf{x}_t$, and an edge direction feature vector $\mathbf{x}_e$ were introduced. A general feature vector $\mathbf{x}$ will be used in this section for discussion purposes only. It should be noted that the discussion applies equally to the feature vectors $\mathbf{x}_c$, $\mathbf{x}_t$, and $\mathbf{x}_e$.

Suppose there are l observations described by a feature vector $\mathbf{x}_i \in R^d$, $i=1,\ldots,l$ and the associated truth $y_i \in \{-1,1\}$. If observation i corresponds to an outdoor image, then $y_i=1$, otherwise $y_i=-1$. The objective is to find some way of separating the observations such that there is a clear distinction between the two classes y_i (i.e. indoor vs. outdoor). There are infinitely many ways of separating the data, however, the separation that yields the best generalization performance is desired. To illustrate this further, a simple linear classification is shown in Figure 6. Figure 6a depicts a successful separation of the data, with minimal margin, where the margin is the sum of the perpendicular distances from the closest point of each class to the separating hyperplane. Figure 6b shows another successful separation where the margin has been maximized.

Intuitively, the case in Figure 6b will have better generalization performance (lower generalization error).

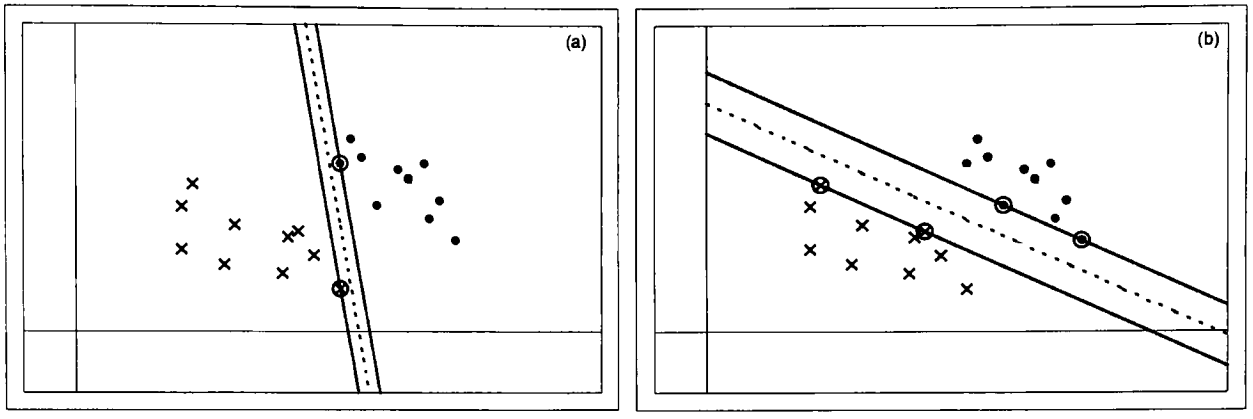


Figure 6. Successful SVM linear classification with sub-optimal (a) and optimal (b) margins.

For the linear-separable case shown in Figure 6, the hyperplane that separates the data satisfies

$$\mathbf{w} \cdot \mathbf{x} + b = 0 \tag{41}$$

Where, $\mathbf{w}$ is normal to the hyperplane, and $|b|/|\mathbf{w}|$ is the perpendicular distance from the hyperplane to the origin. The parameters, $\mathbf{w}$ and b, are determined by training the SVM. The optimal hyperplane is obtained by maximizing the margin subject to the constraints

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1, \text{ for } y_i = +1 \tag{42}$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq -1, \text{ for } y_i = -1 \tag{43}$$

which can be combined into

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0, \forall i \tag{44}$$

All points that satisfy (42) lie on the hyperplane H_1: $\mathbf{x}_i \cdot \mathbf{w} + b = 1$. Points satisfying (43) lie on the hyperplane H_2: $\mathbf{x}_i \cdot \mathbf{w} + b = -1$. Hyperplanes H_1 and H_2 are shown in Figure 6 as solid lines. Any point $\mathbf{x}_i$ lying on either H_1 or H_2 is called a *support vector*. The support vectors in Figure 6

are the points with an additional circle. The perpendicular distance from both H_1 and H_2 to the shattering hyperplane (41) is $1/\|\mathbf{w}\|$, and thus, the margin is simply $2/\|\mathbf{w}\|$.

As stated earlier, the goal is to maximize the margin during SVM training. This optimization problem can be solved using Lagrange multipliers. The full derivation is elegantly laid out in [43, 44]. For the sake of brevity, only the solution is noted below:

$$f(\mathbf{x}) = \sum_{i=1}^{l} \lambda_i y_i \mathbf{x}^T \mathbf{x}_i + b \tag{45}$$

Where, λ_i are the Lagrange multipliers. As can be seen, equation (45) is a function of the observation (or feature) vector, $\mathbf{x}$, and can be interpreted as the distance (in feature space) of the point $\mathbf{x}$ from the separating hyperplane, or decision surface (41).

Up to this point, only the linear, separable SVM case has been treated. A solution similar to (45) can be obtained for a non-linear SVM using a *kernel function* $K(\mathbf{x},\mathbf{x}_i)$. The reader is again referred to [14, 19] for full derivations. The non-linear SVM solution is:

$$f(\mathbf{x}) = \sum_{i=1}^{l} \lambda_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{46}$$

Clearly, Eq. (46) is similar in form to Eq. (45) and in fact, can be said to encompass the linear case, where $K(\mathbf{x},\mathbf{x}_i) = \mathbf{x}^T\mathbf{x}_i$. Some common kernel functions for non-linear classification are listed in Table 1.

Table 1. Possible SVM kernel functions and type of classifier.

Kernel Function	Classifier
$K(\mathbf{x},\mathbf{x}_i) = \exp(-\lvert\mathbf{x}-\mathbf{x}_i\rvert^2 / \sigma^2)$	Gaussian RBF
$K(\mathbf{x},\mathbf{x}_i) = (\mathbf{x}^T\mathbf{x}_i + 1)^p$	Polynomial of degree p
$K(\mathbf{x},\mathbf{x}_i) = \tanh(v(\mathbf{x}^T\mathbf{x}_i) + a)$	Neural Network

Attention is now given to the non-separable data case. Most real life problems are of the non-separable type. The fact that the data is non-separable implies that it is impossible to build a

decision surface without some misclassification. Furthermore, when faced with the non-separable case, the above equations have no solution. To obtain a feasible solution for the non-separable case and to manage the tradeoff between the margin and misclassification, constraints (42) and (43) (for the linear case), are relaxed as follows:

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1-\xi, \text{ for } y_i = +1 \tag{47}$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq -1+\xi, \text{ for } y_i = -1 \tag{48}$$

$$\xi_i \geq 0 \ \forall i \tag{49}$$

For an error to occur ξ_i must exceed unity and hence, $\Sigma\xi_i$ is an upper bound on the number of training errors. A cost parameter C is then introduced, where $C \geq 0$, such that the function to be minimized changes from $\|\mathbf{w}\|^2/2$ to $\|\mathbf{w}\|^2/2 + C\Sigma\xi_i$. The optimization problem can again be solved with Lagrange multipliers, where $0 < \lambda_i < C$. The cost parameter is determined before training by the user; a larger C corresponds to a higher penalty for errors. A similar approach is used for the non-linear SVM. For a more complete description of the solutions incorporating the cost parameter C, the reader is again referred to [45,46].

3.2 Low-Level Feature SVM Training

An RBF kernel (see Table 1) was used to train the color, texture, and edge direction features separately. The choice of kernel was arbitrary. The SVMs were trained using the "SVMfu" algorithm developed at MIT's Artificial Intelligence Lab [50]. The SVMs were trained using low-level features extracted from the full image, as well as image subblocks from a 2 x 2 tessellation and a 4 x 4 tessellation. In each case, the feature vectors were normalized to zero mean and unit variance before training.

3.2.1 *Low-Level Features Extracted From the Full Image*

In this configuration, the color, texture, and edge direction feature vectors $\mathbf{x}_c$, $\mathbf{x}_t$, and $\mathbf{x}_e$ representing the full M x N image were extracted. In general the output of an SVM classifier is a distance measure given by Eq. (46). Hence, the color, texture, and edge direction SVMs output

distance measures $f_c(\mathbf{x}_c)$, $f_t(\mathbf{x}_t)$, and $f_e(\mathbf{x}_e)$ respectively, which represent the indoor/outdoor beliefs for the image in question. The process is shown schematically in Figure 7.

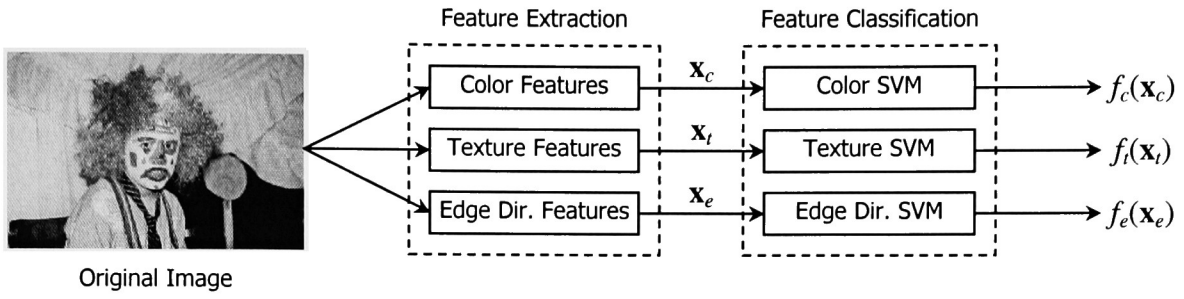


Figure 7. Low-level feature extraction and classification from the full image.

3.2.2 *Low-Level Features Extracted From a 2 x 2 Tessellation*

In this case, the image is divided into 4 subblocks. For a given M x N resolution image, each subblocks will be of size $M/2$ x $N/2$. Let i=1,2,...,4 denote the subblocks of a source image. The feature vectors $\mathbf{x}_c^i$, $\mathbf{x}_w^i$, and $\mathbf{x}_e^i$ are extracted from each subblock i and classified separately. SVM distance measures $f_c(\mathbf{x}_c^i)$, $f_t(\mathbf{x}_t^i)$, and $f_e(\mathbf{x}_e^i)$ are thus, also obtained for each subblock i.

Feature extraction from image subblocks can be expected to achieve less accuracy compared to the full image features, as there are fewer and weaker signatures. Although less accurate, subblock classification offers further alternatives to inferring the final indoor/outdoor classification. For instance, the subblock classification results can be combined in a variety of ways and, as shown in [12], improve the final indoor/outdoor classification. This is the major motivation for exploring subblock classification alternatives. Feature extraction and classification for a 2 x 2 image tessellation is shown in Figure 8.

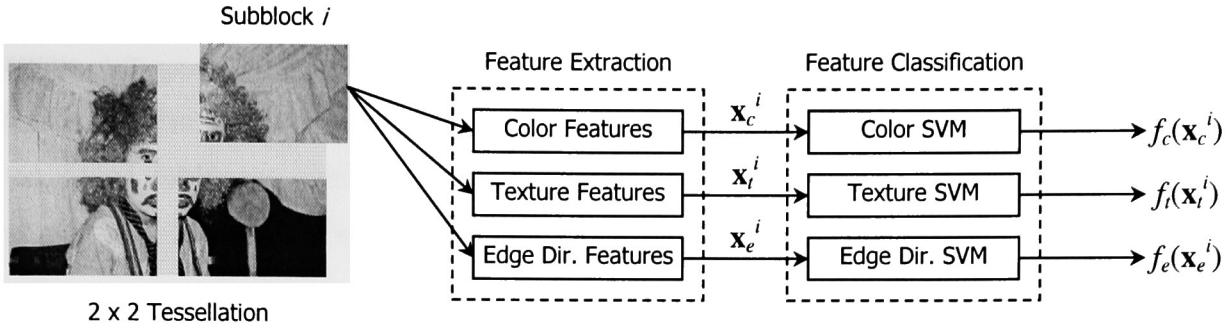


Figure 8. Low-level feature extraction and classification from a 2 x 2 image tessellation.

3.2.3 *Low-Level Features Extracted From a 4 x 4 Tessellation*

A 4 x 4 tessellation of an M x N image results in 16 subblocks of $M/4$ x $N/4$ pixels. As before, the feature vectors $\mathbf{x}_c^i$, $\mathbf{x}_t^i$, and $\mathbf{x}_e^i$, i=1,2,...,16, are extracted from each subblock and classified separately. A further drop in classification rates can be expected from the 4 x 4 as compared to the 2 x 2 tessellation because the subblocks are smaller. Again, SVM distance measures $f_c(\mathbf{x}_c^i)$, $f_t(\mathbf{x}_t^i)$, and $f_e(\mathbf{x}_e^i)$ are obtained for each image subblock. It should be noted that a 4 x 4 tessellation was used in [12].

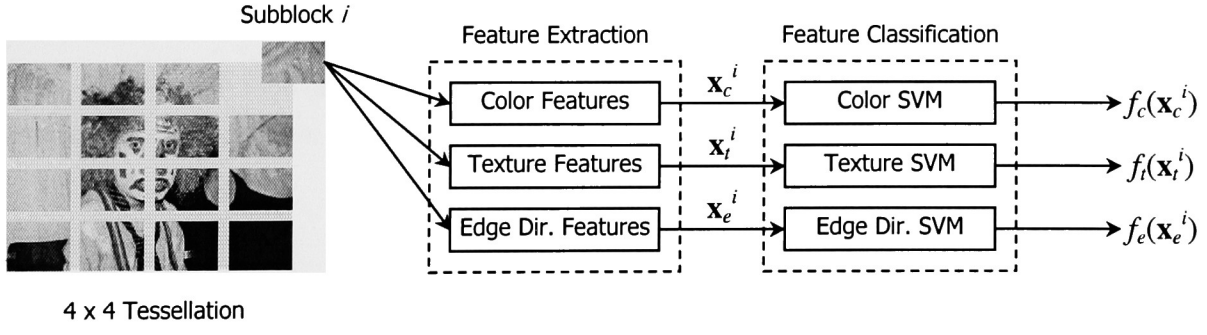


Figure 9. Low-level feature extraction and classification from a 4 x 4 image tessellation.

3.3 Inferring Indoor/Outdoor Classification From Subblocks

As intimated earlier, when low-level features are extracted from the full image, the classifier engine is assigned the task of inferring the high-level indoor/outdoor categorization. Yet, it was

shown in [12] that higher indoor/outdoor classification rates could be achieved by classifying features extracted from image subblocks and then combining the results in a second stage. Specifically, the approach used in [12] involved classifying color and texture features extracted from image subblocks with a k-NN classifier and then combining the subblock results using majority classification to obtain a final indoor/outdoor label for a given image.

Given the above remarks, three approaches to synthesize subblock beliefs were evaluated for the indoor/outdoor classification system proposed here. The first is the majority classification scheme proposed in [12], which will serve as a benchmark.

3.3.1 *Majority Classification*

Assume color, texture, and edge direction feature vectors $\mathbf{x}_c^i$, $\mathbf{x}_t^i$, and $\mathbf{x}_e^i$ are extracted from each subblock i of a given image tessellation. The feature vectors are then classified using the corresponding SVM, in turn producing the distance measures $f_c(\mathbf{x}_c^i)$, $f_t(\mathbf{x}_t^i)$, and $f_e(\mathbf{x}_e^i)$. As described in section 3.1.1, these values measure the distance of a given feature vector from the separating hyperplane—the trained SVM decision boundary in feature space. A large positive value indicates the feature vector has strong outdoor scene cues. Conversely, a large negative value indicates the feature vector has strong indoor scene cues. Thus, *hard* indoor/outdoor labels L_c^i, L_t^i, and L_e^i can be obtained for each subblock i using a *hard limiter*—i.e. thresholding the distance measures at zero:

$$L_c^i = \begin{cases} 1, & f_c(\mathbf{x}_c^i) > 0 \\ 0, & otherwise \end{cases} \tag{50}$$

$$L_t^i = \begin{cases} 1, & f_t(\mathbf{x}_t^i) > 0 \\ 0, & otherwise \end{cases} \tag{51}$$

$$L_e^i = \begin{cases} 1, & f_e(\mathbf{x}_e^i) > 0 \\ 0, & otherwise \end{cases} \tag{52}$$

A label equal to one indicates the subblock represents an outdoor scene. After computing the above equations, each subblock has three indoor/outdoor labels, one for each feature type. Let S represent the summation of labels L_c^i, L_t^i, and L_e^i over all subblocks:

$$S = \sum_{i=1}^{B} L_c^i + L_t^i + L_e^i \tag{53}$$

Where, B is the total number of subblocks (e.g. B=4 for a 2 x 2 tessellation). An indoor/outdoor label can thus be obtained for the whole image according to:

$$L = \begin{cases} 1, & S > 3B/2 \\ 0, & otherwise \end{cases} \tag{54}$$

If, for example, color, texture, and edge direction features are used in a 4 x 4 image tessellation, there will be $3B$ subblock labels. The label L=1 (outdoor) is assigned if the subblock label summation S is greater than half the number of total subblocks; in this case $3B/2$=24. However, not all low-level features need be used. For instance, only two of the low-level features might be extracted and classified. In this case, $2B/2$=16, and therefore, the label L=1 (outdoor) would be assigned if S>16. The majority classifier is shown graphically for a 4 x 4 tessellation in Figure 10.

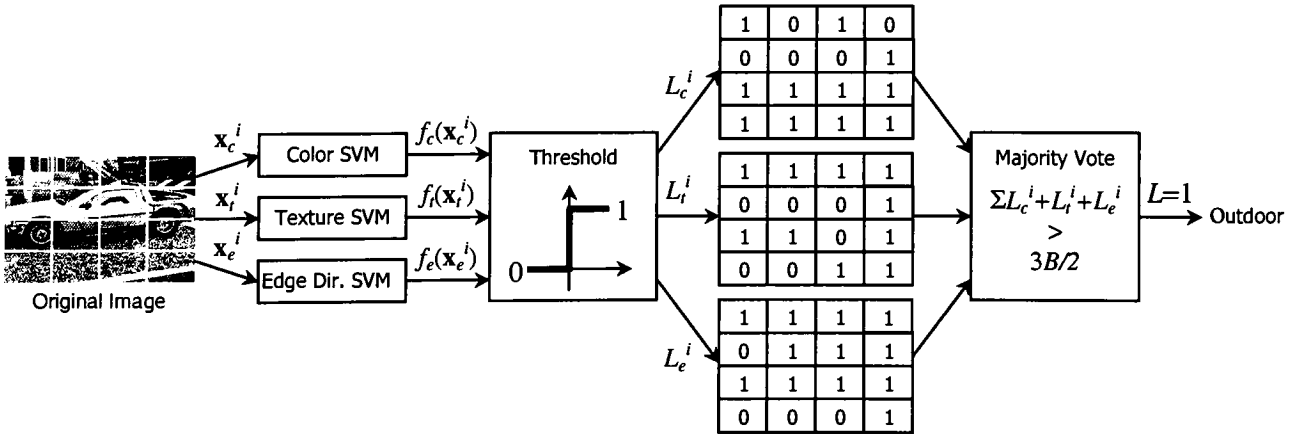


Figure 10. Graphic illustration of the majority classifier approach.

3.3.2 *Synthesis of Subblock SVM Distances*

This approach is very similar to majority classification except that the SVM distance measure is exploited. In the majority classification approach, hard indoor/outdoor labels L_c^i, L_t^i, and L_e^i were obtained by thresholding $f_c(\mathbf{x}_c^i)$, $f_t(\mathbf{x}_t^i)$, and $f_e(\mathbf{x}_e^i)$. Doing so, however, is equivalent to quantizing the distance measures, thus incurring a loss of information and precision. Given that $f_c(\mathbf{x}_c^i)$, $f_t(\mathbf{x}_t^i)$, and $f_e(\mathbf{x}_e^i)$ are distance measures corresponding to all subblocks, they can be summed to obtain a global distance measure for the entire image. In other words, the subblock distance measures can be synthesized to obtain a value that represents the indoor/outdoor belief for the full image. Define three new distance measures corresponding to the entire image:

$$d_c = \sum_{i=1}^{B} f_c(\mathbf{x}_c^i) \tag{55}$$

$$d_t = \sum_{i=1}^{B} f_t(\mathbf{x}_t^i) \tag{56}$$

$$d_e = \sum_{i=1}^{B} f_e(\mathbf{x}_e^i) \tag{57}$$

Where, B again represents the total number of subblocks in the tessellation. A binary indoor/outdoor label can now obtained from the above synthesized distance measures according to:

$$L = \begin{cases} 1, & d_c + d_t + d_e > 0 \\ 0, & otherwise \end{cases} \tag{58}$$

As before, a label L=1 corresponds to an outdoor scene. Summing the subblock distance measures before binarization reduces the impact of any borderline subblocks, as opposed to forcing a hard label. This approach can still be used even if one or more of the low-level features is eliminated. If the edge direction features are excluded, for instance, the label L would be obtained by thresholding the sum of d_c and d_t only. The subblock synthesis approach is shown graphically for a 4 x 4 tessellation in Figure 11.

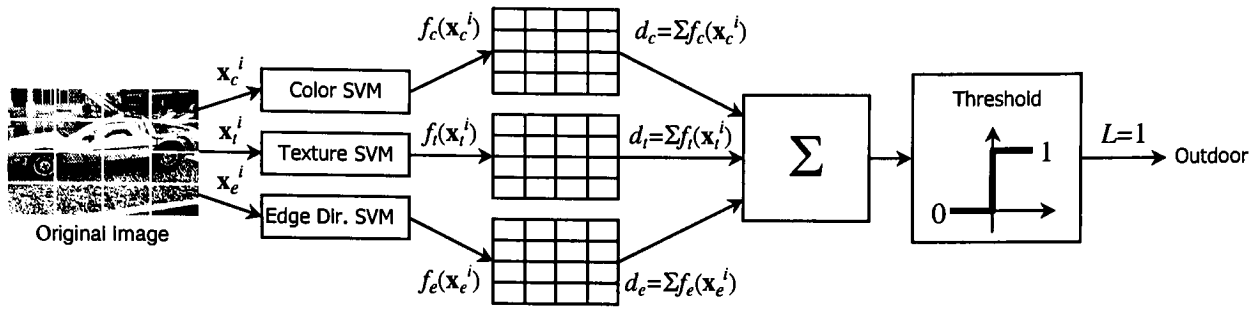


Figure 11. Graphic illustration of the subblock SVM distance synthesis approach.

3.3.3 *Second Stage SVM*

In preceding sections, two approaches were proposed to combine low-level features extracted from image subblocks. In each case, the SVM subblock classification results were combined in distinct ways in order to deduce the indoor/outdoor classification for the full image. A third approach is to use a classifier engine to generalize the subblock classification results and infer a final indoor/outdoor classification for the full image.

In this approach, the subblock SVM distance measures $f_c(\mathbf{x}_c^i)$, $f_t(\mathbf{x}_t^i)$, and $f_e(\mathbf{x}_e^i)$ are synthesized as in section 3.3.2 in order to obtain distance measures d_c, d_t, and d_e using Eq. (55) – (57). These distance measures are then used to form a new color, texture, and edge direction feature vector $\mathbf{x}_{cte}$:

$$\mathbf{x}_{cte} = [d_c, d_t, d_e] \tag{59}$$

A new RBF SVM is then trained using the feature vector $\mathbf{x}_{cte}$. The output of this SVM, $f_{cte}(\mathbf{x}_{cte})$, thus represents an indoor/outdoor belief for the entire image. It can be said that the overall process is a two-stage approach. The first stage involves extraction and classification of the low-level subblock features $\mathbf{x}_c^i$, $\mathbf{x}_t^i$, and $\mathbf{x}_e^i$. The second stage involves synthesizing the subblock SVM distance measures $f_c(\mathbf{x}_c^i)$, $f_t(\mathbf{x}_t^i)$, and $f_e(\mathbf{x}_e^i)$ in order to obtain the full image color, texture, and edge direction features d_c, d_t, and d_e. These features are then classified using a newly trained second

stage SVM in order to obtain the final indoor/outdoor classification. The two-stage SVM approach is shown graphically for a 4 x 4 tessellation in Figure 12.

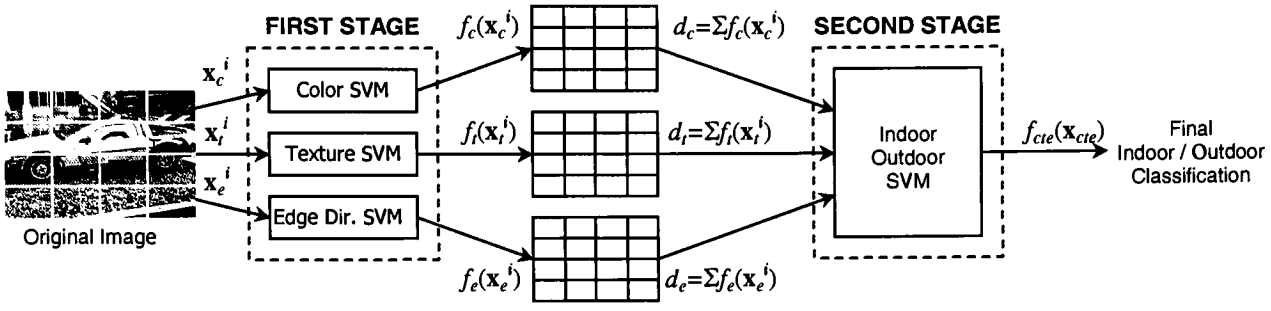


Figure 12. Graphic illustration of the two-stage SVM approach.

4. SEMANTIC FEATURE EXTRACTION

Although classifier engines can be used to establish a relationship between image primitives and semantic scene understanding (e.g. indoor vs. outdoor), the approach can be enriched by incorporating additional knowledge that is pertinent to the semantic scene understanding in question. In this case, the semantic scene understanding in question is whether or not a particular scene is indoor vs. outdoor. Additional knowledge that is pertinent to this task might be whether or not the scene contains grass and/or sky regions, for example. If the scene does contain grass and/or sky, then it can be asserted that it is an outdoor scene. This assertion is not categorical, as there are ambiguous cases such as photographs taken through windows. However, these cases are infrequent and generally without resolution involving philosophical discourse. Hence, it is reasonable to assume that additional knowledge of the scene might reinforce the indoor/outdoor categorization obtained using low-level image analysis. Two inevitable questions arise. Can additional knowledge of the scene be obtained reliably? And if so, how can this knowledge be incorporated with the aforementioned low-level features?

For the application considered here, semantic scene content such as grass, sky, buildings, cars and people can be said to represent *mid-level* scene information in that it is less general than the indoor vs. outdoor labeling. Prior image understanding research has shown that such mid-level scene content can be detected reliably. Some examples include vegetation detection [51], sky detection [51,52], and people detection [53]. Furthermore, models for probabilistic integration of scene information also exist. Specifically, the use of Bayesian networks for feature integration is discussed in section 5.

Not all mid-level scene information is useful in determining whether or not a given image is an indoor scene or an outdoor scene. For example, people can be present in both indoor and outdoor scenes. However, only on rare occasions can grass be found indoors (e.g. a domed stadium). Similarly, sky regions are almost always present in outdoor scenes. Given this reasoning, the presence of sky and grass mid-level information for improved indoor/outdoor classification is considered here.

To propose a scheme for the detection of grass and sky in images is beyond the scope of this work. Instead, reliable grass and sky ground truth associated with an image database provided by Kodak (see section 6.1) was used. The sky ground truth is further qualified as blue sky, cloud, mixed sky, twilight, or other sky. The use of ground truth provides an upper bound on the indoor/outdoor classification accuracy. To ascertain how accurate the indoor/outdoor classification might be with computed mid-level information, grass and sky detection schemes developed by Kodak were used. Kodak's sky detection algorithm is as described in [52]. Though undocumented, the grass detection algorithm employs color and texture information to detect grass regions. An example image and its associated grass ground truth are shown in Figure 13.

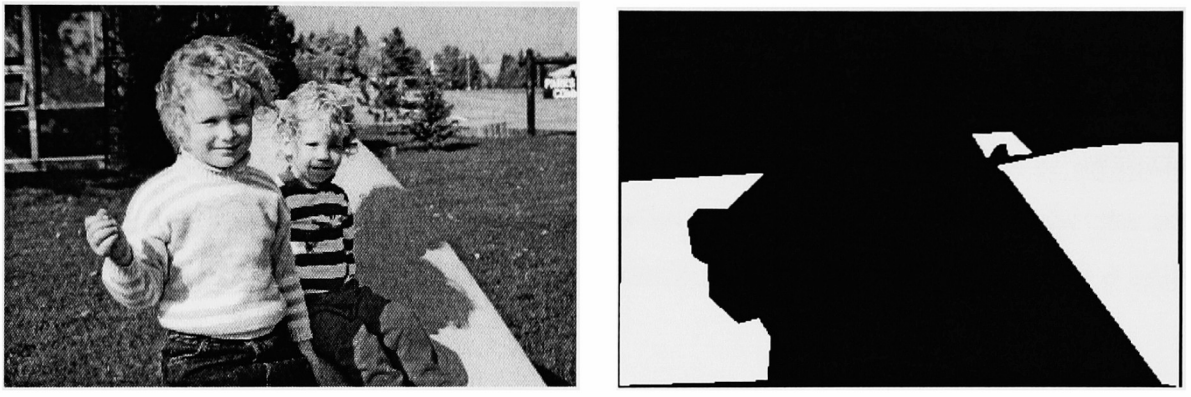


Figure 13. Example image (a) and associated grass region ground truth (b).

5. FEATURE INTEGRATION

In this section, the process of integrating low-level features (as those described in section 2) and semantic features (as those described in section 4) for enhanced indoor/outdoor classification is discussed. An introduction to Bayesian networks—probabilistic inference engines—is first provided.

5.1 Bayesian Networks

Bayesian networks, also known as belief networks, or simply Bayes nets, provide a powerful framework for the description of complicated probabilistic systems through simple conditional relationships [54]. They have become an important tool in the field of artificial intelligence, which is ruled by uncertainty. Bayes' theorem is one of the celebrated results of probability theory. It states that the posterior (or *a posteriori*) probability is described by the joint probability, which in turn, is described by the conditional probability and the prior (or *a priori*) probability:

$$P(H \mid E) = \frac{P(H,E)}{P(E)} = \frac{P(E \mid H)P(H)}{P(E)} \tag{60}$$

The latter part of Eq. (60) is the well-known inversion formula. In words, it states that the belief hypothesis H is true, based on new evidence E (posterior probability), can be expressed by the product of the previous belief H is true (prior probability) with the likelihood that E will occur if H is true (conditional probability).

The importance of this result is that $P(H \mid E)$, a typically difficult quantity to assess, can be obtained from quantities that are not only more accessible, but usually available from experiential knowledge. Yet, often the evidence E is not a single variable but rather, a set of variables. As the number of evidence variables increases, computation of the joint probability becomes intractable. Furthermore, it has been observed that a purely mathematical description of probabilistic reasoning is devoid of psychological meaning and thus differs from human probabilistic

reasoning. Perhaps the most striking limitation of numerical approaches to probability is the assessment of independence. Using the previous example, if the hypothesis H is independent of the evidence E, then,

$$P(H, E) = P(H)\, P(E) \tag{61}$$

And thus, the posterior probability is equal to the prior probability,

$$P(H \mid E) = \frac{P(H, E)}{P(E)} = P(H) \tag{62}$$

In practice, independence is gauged by computing the product $P(H)P(E)$ and determining whether or not it is equal to the joint probability. Although more formal, it is impractical and again deviates from human intuition. In fact, humans are quickly and confidently able to determine independence without computing numerical probabilities. Similarly, the notion of conditional independence is quite familiar to humans. For example, assume there are two distinct evidence variables, E_1 and E_2. It can be said that the hypothesis H is independent of E_2, given E_1 if

$$P(H \mid E_2, E_1) = P(H \mid E_1) \tag{63}$$

Although H and E_2 may be marginally dependent, they become independent when E_1 is known—i.e. conditionally independent. In other words, E_2 is rendered irrelevant given knowledge of E_1. Humans are able to confidently deduce this sort of relevance from the structure of human memory, which is far more efficient than assessing dependence via numerical estimates. Thus, there is motivation for the use of a probabilistic architecture that is closer to human reasoning and exploits conditional independence for added simplicity. Bayesian networks provide such a framework.

Bayesian networks are directed acyclic graphs (DAG), where the nodes represent variables and the links between nodes represent causal dependence expressed by conditional probabilities. A link originates at a *parent* node and is directed toward a *child* node. The direction of the link

indicates causality, and thus a dependence relationship. Nodes that exist at the same level are considered conditionally independent. Such a framework can be regarded as a knowledge representation because it encodes the joint probability of the variables. Also, computation of the joint probability, which as described earlier, can become intractable with a large number of variables, is simplified by taking advantage of the conditional independence between variables.

In the context of semantic scene understanding, the nodes in a Bayesian network represent features. These features may describe low-level or high-level scene information. A Bayesian network has four components: 1) prior belief about the features; 2) conditional probability matrices (CPMs) that describe the relationship between connected nodes; 3) evidence from feature detectors that are supplied as input; and 4) posterior belief after the conditional probabilities are propagated through the Bayesian network.

Training Bayesian networks involves determining the CPMs for each parent-child node relationship. This is facilitated by the fact that links at the same level are considered independent. Two methods have been used to obtain CPMs. The first is via expert knowledge—an *ad hoc* approach—where an expert provides information regarding the conditional probabilities of each feature detector. The second is through contingency tables, where observations of each feature detector are recorded and compiled using sampling and correlation methods. The contingency table can then be normalized and used as the CPM. The CPM associated with a link cannot be trained using frequency counting unless ground truth is available.

Bayesian networks have been shown to useful in data fusion applications [55] and have also been successfully employed for semantic scene understanding [15,56,57]. The work of Luo and Savakis [15] is worthy of mention as it applied Bayesian Networks to the indoor/outdoor classification problem. Given that the work of [15] involved the same image data set used here, the same Bayesian network structure can be applied to the low-level feature-based indoor/outdoor classification system presented in previous sections.

5.2 Indoor/Outdoor Feature Integration

Integration of the low-level features described in section 2, together with the grass and sky semantic features of section 4 is accomplished using a Bayesian network. The output of the Bayesian network indicates the probability that the image in question is an outdoor scene. The Bayesian network structure is shown in Figure 14.

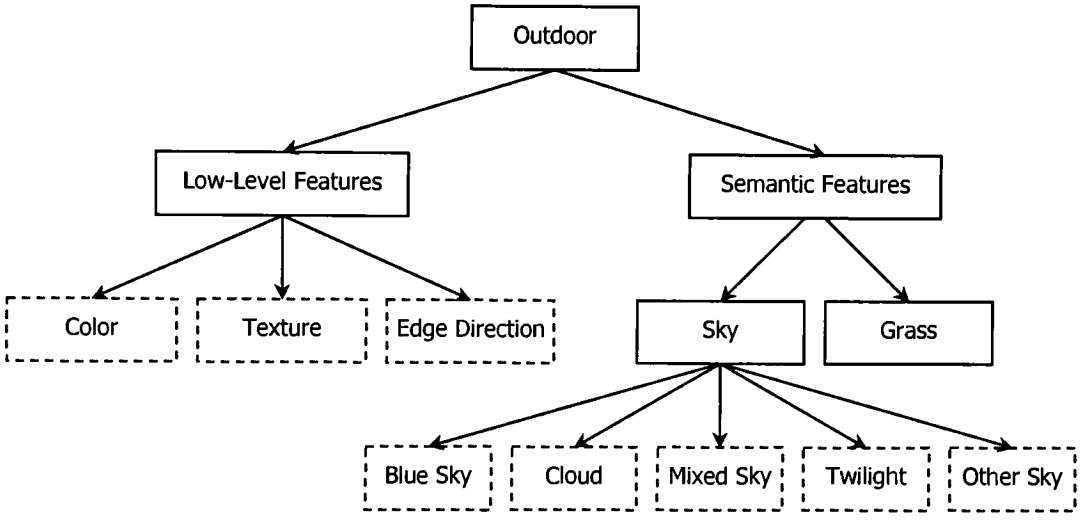


Figure 14. Bayesian network for indoor/outdoor classification.

The Bayesian network was applied in two different ways. First, the *Color*, *Texture*, and *Edge Direction* nodes (shown with dashed lines in Figure 14) were pruned and the indoor/outdoor belief $f_{cte}(\mathbf{x}_{cte})$ obtained from the second stage SVM (section 3.3.3) was used to represent the *Low-Level Features* node directly. Because the $f_{cte}(\mathbf{x}_{cte})$ represents a distance measure, theoretically, its values can range from negative infinity to positive infinity:

$$-\infty < f_{cte}(\mathbf{x}_{cte}) < \infty \tag{64}$$

Thus, for inclusion in the probabilistic Bayesian network, $f_{cte}(\mathbf{x}_{cte})$ was translated to a valid probability distribution. This was done using the sigmoid function:

$$\Pr\{\text{Outdoor} \mid \text{Low-Level Features}\} = \frac{1}{1+\exp[-f_{cte}(\mathbf{x}_{cte})]} \tag{65}$$

Where, Pr{Outdoor | Low-Level Features} $\in$ [0,1] represents the probability that an image is an outdoor scene, given the low-level feature classification of section 3.3.3.

In the second approach, the *Color*, *Texture*, and *Edge direction* nodes were retained and the color, texture, and edge direction beliefs obtained from the separate SVM classifiers were used as inputs. In this configuration, the Bayesian network infers the belief for the *Low-Level Features* node. Comparing these two approaches will determine whether the second stage SVM or the Bayes net does a better job at inferring the indoor/outdoor classification from low-level features.

The semantic feature probabilities were obtained using ground truth. That is, for each image in the image database used here, there is associated ground truth indicating the presence of grass, blue sky, clouds, mixed sky, twilight, or other sky. In addition, computed grass and sky probabilities, obtained for each image in the database using Kodak's detectors, were also evaluated. It should be noted though, that Kodak's sky detector does not discriminate between different types of sky; it merely yields the probability that a scene contains sky of any type. Hence, when using computed semantic features, the *Blue Sky, Cloud, Mixed Sky, Twilight*, and *Other Sky* nodes (shown with dashed lines in Figure 14) were pruned from the Bayes net.

6. RESULTS AND DISCUSSION

6.1 Image Database

A database of 1200 consumer photographs collected by Kodak was used to train and test the indoor/outdoor classification performance. It is the same image database as the one used in [12,15]. However, the number of images was reduced from 1343 to 1200 by eliminating images with near duplicate scene content and/or ambiguous indoor/outdoor labeling. The removal of near duplicates can, in general, result in higher error rates, as will be shown later. The indoor and outdoor images are equally distributed in the 1200 image set.

The images in the database are 36-bit color, 512 x 768 resolution scanned photographs. The preprocessing stage included quantization to 24-bit color, and a simple color balance that clipped the top and bottom 0.5% of each color channel, centered, and equalized the histogram. In addition, the images were subsampled to 256 x 384 pixels for increased processing speed. For training and testing, the image database was divided into two independent sets of 600 images. If not explicitly stated, the indoor/outdoor classification rates reported in the following sections correspond to the independent test set.

6.2 Low-Level Feature Classification

As described, earlier, color histograms, wavelet texture features, and edge direction histograms (section 2) were extracted in a variety of arrangements (section 3). The following sections describe the classification results.

6.2.1 *Low-Level Features Extracted From the Full Image*

In this configuration, feature vectors $\mathbf{x}_c$, $\mathbf{x}_t$, and $\mathbf{x}_e$ were extracted from the full image and classified using separately trained SVMs. Training the features separately provides insight into

the discriminatory power of each type of feature, as they describe different image characteristics. The results are shown in Table 2.

Table 2. Classification results for low-level features extracted from the full image.

Feature	Training Set	Test Set
Color	78.8%	74.5%
Texture	84.5%	83.0%
Edge Direction	84.0%	72.5%

In comparing the results of Table 2 with prior indoor/outdoor classification work, the results are quite favorable. In terms of the color histograms, an indoor/outdoor classification accuracy of 74% was reported in both [12] and [15]. As can be seen from Table 2, an indoor/outdoor classification accuracy of 74.5% was obtained here using SVMs. The same image database was used in [12,15], which suggests that the comparable accuracy—despite the reduced feature dimensionality—can be attributed to the SVM's superior generalization ability compared to the *k*-NN classifier of [12,15].

The wavelet texture features also achieved good results. The 83.0% classification rate shown in Table 2 is higher than the MSAR results of 82.2% reported in [12,15]. This is a notable result given that the wavelet texture features described in section 2.2 are more computationally efficient and have half the dimensionality of the MSAR features.

Similarly, the edge direction histograms also performed well. In [11,14], it was reported that edge direction histograms (with n_e=72) extracted from the full image achieve an indoor/outdoor classification accuracy of approximately 60% (on a distinct image database). Clearly, the classification accuracy reported here is considerably higher; again with half the feature dimensionality.

6.2.2 *Low-Level Features Extracted From a 2 × 2 Tessellation*

In sections 3.2.2 and 3.2.3 it was discussed how low-level features could be extracted from image subblocks. Color, texture, and edge direction feature vectors $\mathbf{x}_c^i$, $\mathbf{x}_t^i$, and $\mathbf{x}_e^i$ were extracted from

each subblock i, where i=1,2,...,B and B=4 for a 2 x 2 tessellation. The classification results are shown in Table 3.

Table 3. Classification results for low-level features extracted from a 2 x 2 tessellation.

Feature	Training Set	Test Set
Color	78.3%	70.1%
Texture	80.1%	79.0%
Edge Direction	67.3%	64.4%

As expected, the results of Table 3 show a decrease compared to Table 2. Aside from the overall drop in accuracy compared to Table 2, it is interesting to note that the edge direction features showed a particular decrease in accuracy. This implies that there are insufficient edges with dominant orientations in image subblocks. The figures in Table 3 cannot be compared to other approaches, as no prior work in indoor/outdoor classification has reported results from a 2 x 2 tessellation. As discussed earlier, the main motivation in considering image subblocks is that the results can then be combined. Unfortunately, because a 2 x 2 tessellation only has four subblock, there is only small number of samples to combine. Therefore, focus is instead placed on a 4 x 4 tessellation, which was used to great effect in [12].

6.2.3 *Low-Level Features Extracted From a 4 x 4 Tessellation*

In this approach, color, texture, and edge direction feature vectors $\mathbf{x}_c^i$, $\mathbf{x}_t^i$, and $\mathbf{x}_e^i$ were extracted from each subblock i, where i=1,2,...,B and B=16 for a 4 x 4 tessellation. The classification results are shown in Table 4.

Table 4. Classification results for low-level features extracted from a 4 x 4 tessellation.

Feature	Training Set	Test Set
Color	73.7%	67.6%
Texture	75.8%	73.0%
Edge Direction	65.6%	57.7%

The results in Table 4 reflect a further drop in accuracy with smaller subblocks. In particular, the classification accuracy using edge direction histograms is very low. The 57.7% accuracy is

barely better than simply venturing a guess. Though low, the color and texture results do compare positively with previous work. Using a 4 x 4 tessellation on the same image set, a classification accuracy of 70.3% for color features and 74.7% for MSAR features were reported in [12]. The results of Table 4 are only slightly lower, which indicates that positive results can be achieved with reduced feature dimensionality and improved computational efficiency.

6.3 Inferring Indoor/Outdoor Classification From Subblocks

Although classification accuracy is lower for features extracted from image subblocks compared to full image features, they provide a richer description of the scene. Their usefulness lies in the fact that the subblock classification results can be combined in order to obtain higher indoor/outdoor classification rates than from low-level features extracted and classified for the full image, as demonstrated in [12].

6.3.1 *Majority Classification*

The majority classification approach was evaluated because it was used successfully for indoor/outdoor classification in [12]. The results using a majority classifier with different combinations of color, texture, and edge direction features are shown in Tables 5 and 6. Table 5 shows the results using a 2 x 2 tessellation and Table 6 shows the results using a 4 x 4 tessellation.

Table 5. Indoor/Outdoor classification results using a majority classifier (2 x 2 tessellation).

Feature Combination	Indoor/Outdoor Classification Rate
Color	73.2%
Texture	82.2%
Edge Direction	70.7%
Color and Texture	85.7%
Color and Edge Direction	77.7%
Texture and Edge Direction	80.2%
Color, Texture, and Edge Direction	85.7%

Table 6. Indoor/Outdoor classification results using a majority classifier (4 x 4 tessellation).

Feature Combination	Indoor/Outdoor Classification Rate
Color	83.8%
Texture	81.0%
Edge Direction	70.0%
Color and Texture	87.2%
Color and Edge Direction	86.0%
Texture and Edge Direction	79.0%
Color, Texture, and Edge Direction	87.0%

In comparing Tables 5 and 6, it is clear that the indoor/outdoor classification results using a 4 x 4 tessellation are better than the results using a 2 x 2 tessellation. This suggests that a larger number of subblocks provide a more detailed description of the indoor/outdoor scene layout. Also, the classification rates shown in Tables 5 and 6 are higher than the results obtained using full image low-level feature extraction and classification (Table 2). This confirms the notion that, when combined, the subblock classification results provide a richer depiction of the scene than full image features.

Another observation is that the edge direction features performed poorly in both Tables 5 and 6. As can be discerned from Table 6, the highest indoor/outdoor classification rate (87.2%) was achieved using the combination of color and texture features. There was no added advantage to combining edge direction features together with the color and texture features. In fact, the edge direction features in combination with the texture features actually result in a lower classification rate than the texture features alone.

The 87.2% classification rate of Table 6 compares well with existing indoor/outdoor classification approaches. Using a majority classifier on a 4 x 4 tessellation, Szummer and Picard [12] reported a classification accuracy of 90.3% with virtually the same imagery used here. However, as noted in section 6.1, the database contained many duplicate scenes, which can lead to higher classification rates. In fact, an informal reassessment of Szummer and Picard's approach after removing images with duplicate scene content showed that the overall classification rate dropped to about 85%. Therefore, the results of Table 6 are comparable, if not superior to those of [12]. The indoor/outdoor classification rate of 87.2% is still 1 to 2 percent

lower than the 88.2% and 88.7% reported in [11,14]. For this reason, other approaches to synthesize subblock classification results were explored.

6.3.2 *Synthesis of Subblock SVM Distances*

It was noted in section 3.3.2 that a majority classifier does not take full advantage of the SVM distance measure. On the other hand, the results in Tables 7 and 8 illustrate that the method of 3.3.2 exploits the subblock SVM distance measures for added classification accuracy, as shown below.

Table 7. Indoor/Outdoor classification results using subblock synthesis (2 x 2 tessellation).

Feature Combination	Indoor/Outdoor Classification Rate
Color	74.7%
Texture	86.2%
Edge Direction	72.0%
Color and Texture	88.5%
Color and Edge Direction	80.5%
Texture and Edge Direction	85.7%
Color, Texture, and Edge Direction	88.5%

Table 8. Indoor/Outdoor classification results using subblock synthesis (4 x 4 tessellation).

Feature Combination	Indoor/Outdoor Classification Rate
Color	85.5%
Texture	85.0%
Edge Direction	76.7%
Color and Texture	89.0%
Color and Edge Direction	88.5%
Texture and Edge Direction	85.3%
Color, Texture, and Edge Direction	88.0%

The highest indoor/outdoor classification rate (89.0%) was obtained by combining color and texture features from the 4 x 4 tessellation (Table 8). The edge direction features again performed poorly, adding no value to most of the combinations shown in Tables 7 and 8. The indoor/outdoor classification rate of 89.0% (Table 8) is about 2 percent better than the 87.2% obtained using a majority classifier on a 4 x 4 tessellation (Table 6). This is a notable increase in accuracy considering such a small change in approach. The benefit is most likely due to the fact

that this approach minimizes the impact of subblocks with a borderline categorization. For example, if a given subblock has confusing signatures and could be classified as either indoor or outdoor, its SVM distance measures $f_c(\mathbf{x}_c^i)$, $f_t(\mathbf{x}_t^i)$, and $f_e(\mathbf{x}_e^i)$ will all be near zero. Therefore, the effect is minimal in summing the SVM distance measures over the other subblocks in the image. On the other hand, the majority classifier forces and indoor or outdoor label potentially amplifying the ambiguity and incurring a classification error.

6.3.3 *Second Stage SVM*

In this approach, an SVM classifier is used to infer the indoor/outdoor classification from the subblock beliefs. Given the poor performance of the edge direction features shown in previous sections, they were eliminated from consideration for the second stage SVM training. Because the 4 x 4 tessellation results were higher than the 2 x 2 tessellation results in each of the approaches discussed up to this point, the 2 x 2 configuration was also discarded. Thus, the following feature vector was used for the second SVM classification stage instead:

$$\mathbf{x}_{ct} = [d_c, d_t] \tag{66}$$

Where, d_c and d_t are color and texture distance features obtained using Eq. (55) and (56) from a 4 x 4 tessellation. Using the new feature vector, a new SVM was trained on the image set. The second stage SVM results for both the training and test sets are shown in Table 9.

Table 9. Indoor/Outdoor classification results using a second stage SVM.

Training Set	Test Set
95.0%	90.2%

As can be seen from Table 9, the second stage SVM classification yields a 1% accuracy increase compared to Table 8 and a 3% increase compared to Table 6. The increase is not unexpected, as classifier engines are able to exploit non-linear interaction between features.

In comparing the results of Table 9 with prior research, the indoor/outdoor classification accuracy of 90.2% is comparable, if not superior to existing methods. It was already mentioned that

Szummer and Picard's method [12] actually achieved a classification of about 85% when duplicate scenes were removed from Kodak's image database. Similarly, Paek et al [13] reported an indoor/outdoor classification rate of 86% on a database of 1300 news images. Therefore, the second stage SVM approach described here represents an increase of roughly 5% compared to those two methods. In [11,14], Vailaya et al obtained indoor/outdoor classification rates of roughly 88.2% and 88.7% on two different test sets. Again, these figures are below those of Table 9—by as much as 2%. Finally, Luo and Savakis [15] reported an indoor/outdoor classification accuracy of 90.1% and 84.7% by integrating low-level features with ground truth and computed semantic features (respectively) using a Bayes net. A similar approach was also used here (section 5) and the results are presented in the following sections.

6.4 Feature Integration Using a Bayesian Network

A similar Bayes net approach to integrating low-level and semantic features was also discussed in [15]. However, the low-level features used in [15] were the same as those used in [12], which as discussed earlier did not perform as well as the color and texture features proposed here. Thus, it is reasonable to assume that a similar integration of low-level and semantic features might further increase the accuracy of the indoor/outdoor classification system proposed here. The first approach discussed here involved using the second stage SVM to combine the color and texture features and then use the Bayes net to integrate the results with the semantic features. The second approach involved using the Bayes net to combine the color and texture features directly and then integrate them with the semantic features.

6.4.1 *Bayesian Network Feature Integration Using Second Stage SVM results*

In this case, the *Color*, *Texture*, and *Edge Direction* nodes of the Bayes net (Figure 14) were pruned and the second stage SVM results were used as direct input to the *Low-Level Features* node (Figure 14). In such a scenario, the second stage SVM combines the color and texture features in order to infer a low-level feature-based indoor/outdoor probability, as in Eq. (61). The

results using Kodak's grass and sky detectors to compute the semantic feature probabilities are shown in Table 10. The results using grass and sky ground truth are shown in Table 11.

Table 10. Results using the second stage SVM belief with computed semantic features.

	Correct	Incorrect	Percent Correct
Indoor	266	30	89.9%
Outdoor	277	27	91.1%
Overall	543	57	90.5%

Table 11. Results using the second stage SVM belief with ground truth semantic features.

	Correct	Incorrect	Percent Correct
Indoor	275	21	92.9%
Outdoor	278	26	91.4%
Overall	553	47	92.2%

There are some important observations to be made from Tables 10 and 11. Table 10 shows an overall indoor/outdoor classification rate of 90.5% when combining low-level features and semantic features. The first observation to be made is that this figure is essentially the same as the indoor/outdoor classification rate of 90.2% obtained using the second stage SVM approach, as shown in Table 9. In other words, there was not a significant change in classification accuracy when adding semantic scene knowledge. This is somewhat surprising, as one would expect knowledge of grass and/or sky in a scene to aid indoor/outdoor classification. Ideally, the grass and sky detection should help increase the indoor/outdoor classification by eliminating false positives and true negatives. The fact that the indoor/outdoor classification rate did not increase notably may be because any gains obtained through correct grass and sky detection were cancelled out by errors due to incorrect grass and sky detection. In other words, for every outdoor scene with correctly detected grass and sky, there may have been an indoor scene with incorrectly detected grass and sky. Moreover, it is possible that scenes incorrectly classified by the second stage SVM did not contain either grass or sky, in which case the semantic features would be of no aid. This is confirmed by the incorrectly classified images shown in Figure 15. Figure 15a shows an outdoor scene incorrectly classified as an indoor scene while Figure15b shows an indoor scene incorrectly classified as an outdoor scene.

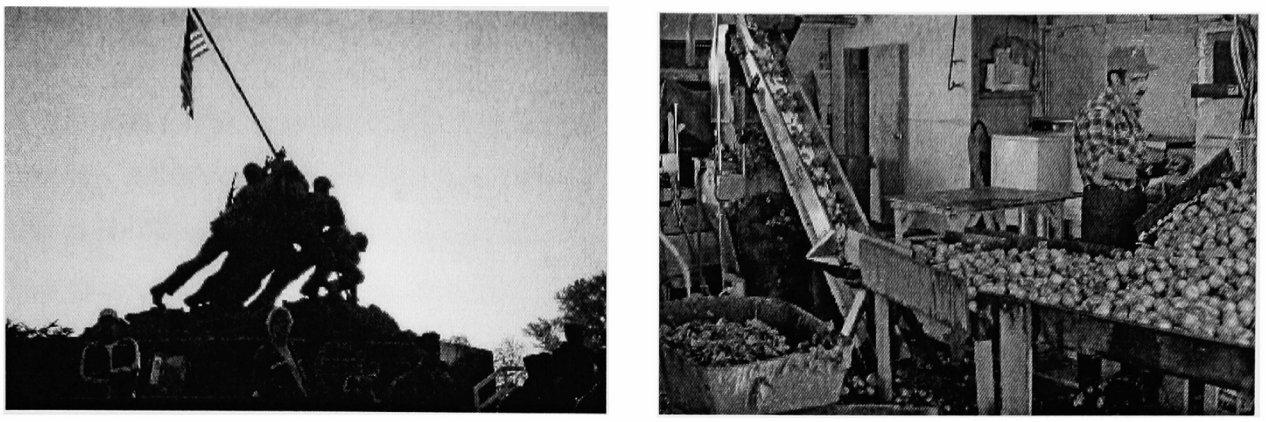


Figure 15. Incorrectly classified outdoor scene (a) and indoor scene (b).

As can be seen, the image in Figure 15a contains twilight sky, which is not explicitly detected by Kodak's sky detector. The image in Figure 15b was probably incorrectly classified because the natural light and the texture are more indicative of an outdoor scene. It is possible that sky ground truth, where twilight sky is considered, might benefit a case like the image in Figure 15a.

Table 11 shows the potential impact of incorporating semantic features assuming perfect grass and sky detection. A full 2% increase in classification accuracy is obtained compared to the results of Table 10. Of particular note is the increase in correctly classified indoor scenes. The number of correctly classified indoor scenes increased from 266 in Table 10 to 275 in Table 11, while the number of correctly classified outdoor scenes remained essentially constant. This suggests that accurate knowledge of the presence of sky and grass regions in an image helps eliminate false positives, i.e. indoor scenes classified as outdoor scenes.

Although there was a notable increase in classification accuracy using grass and sky ground truth, it may be worthwhile to include additional semantic features other than sky and grass. It is possible that the indoor/outdoor color and texture features are inherently using sky and grass to distinguish between indoor and outdoor scenes. Thus, inclusion of semantic features not reliably described by color and texture low-level features might increase indoor/outdoor classification using computed semantics.

6.4.2 *Full Bayesian Network Feature Integration*

A full Bayes net approach to low-level and semantic feature integration is discussed in this section. In this case, the *Color* and *Texture* nodes of the Bayes net (Figure 14) were retained. The *Edge Direction* node was pruned, however, due to the poor performance of the edge direction histograms shown in previous sections. The color and texture SVM distance measures d_c and d_t were converted to probabilities using a sigmoid function akin to Eq. (65). Such an approach yielded color and texture indoor/outdoor classification rates of 85.5% and 85.0%, respectively (see Table 8). The Bayes net is then used to integrate the low-level and semantic features. As in the previous section, both computed and semantic grass/sky features were used.

Table 12. Results using Bayes net to integrate low-level and computed semantic features.

	Correct	Incorrect	Percent Correct
Indoor	260	36	87.8%
Outdoor	284	20	93.4%
Overall	544	56	90.7%

Table 13. Results using Bayes net to integrate low-level and ground truth semantic features.

	Correct	Incorrect	Percent Correct
Indoor	277	19	93.6%
Outdoor	280	24	92.1%
Overall	557	43	92.8%

The trends in Tables 12 and 13 are similar to those observed in Tables 10 and 11 in the previous section. Again, the overall classification rate of 90.7% using both low-level and computed semantic features (Table 12) is not significantly higher than the classification rate of 90.2% using low-level features exclusively (Table 9).

The potential gains in accuracy when incorporating semantic features can be seen in Table 13. The use of grass and sky ground truth increased the overall indoor/outdoor classification rate to 92.8%, which is almost a full 3% improvement compared to the second stage SVM classification rate of 90.2% (Table 9). It is also interesting to note that the results of Tables 12 and 13 are slightly higher than the results of Tables 10 and 11. This implies that the Bayes net does a better job of combining the color and texture features compared to the second stage SVM.

6.5 Computational Efficiency

One of the goals in developing the indoor/outdoor classification system proposed here was to address the issue of computational efficiency. The first gain in computational efficiency was to reduce the dimensionality of the feature set below the norm for similar applications. As discussed in section 2, the proposed color, texture, and edge direction features possess half the number of dimensions compared to existing methods. This is an important improvement, as it is well known that very high-dimensional feature sets can make training and classification intractable.

Another important gain is the reduced computational complexity of the low-level features—especially the texture features. Prior approaches to indoor/outdoor classification employed the popular MSAR texture features [11,12,14,15]. The wavelet texture features introduced here are a more computationally viable alternative to the MSAR model. To highlight this point, computation (on a Sun Ultra 5) of the MSAR features on an image of comparable size to those in our database required 194 seconds compared to only 0.3 seconds for the wavelet features. Moreover, computation of simple color histogram features (as the ones used here) is trivial and much more efficient than say, the color moments used in [11,14,17]. Most importantly, however, the gains in low-level feature computational efficiency did not compromise the classification accuracy of the system.

In terms of low-level feature classification, SVMs were used instead of the notoriously slow *k*-NN classifier used in [12,15,16,17]. Whereas a *k*-NN classifier must parse the entire training space (equal to the number of training samples) to classify a given feature vector, the number of points in the SVM training space is equal to the number of support vectors [45] (typically less than the number of training samples). In fact, the color and texture SVMs ultimately used for indoor/outdoor classification here, represent a combined 33% decrease in training vectors compared to a *k*-NN classifier.

Finally, efficient integration of low-level and semantic features is afforded by the use of a Bayesian network. Belief networks provide fast and efficient computation of potentially complicated probabilistic systems by reducing the relationships in the system to known conditional dependencies.

7. CONCLUSIONS

It was shown that a set of low dimensional, computationally efficient low-level features could be used to accurately classify indoor and outdoor scenes. In particular, color histograms extracted from the *LST* color space and wavelet texture features were shown to be useful. SVMs were used for enhanced classification performance of the reduced dimensionality feature set. It was also demonstrated that combining classification results from a 4 x 4 tessellation led to higher indoor/classification accuracy than features extracted using other configurations. Using the combined subblock results, an indoor/outdoor classification accuracy of 90.2% was obtained using a second stage SVM classification. This is an encouraging result, as it represents a 2 to 5 percent increase in accuracy compared to previous low-level feature approaches to indoor/outdoor classification [11,12,13,14,16]. Finally, incorporation of semantic features using a Bayesian network was shown to provide a potential gain of 3% in indoor/outdoor classification accuracy, raising the overall rate to 92.8%.

Given that the incorporation of computed semantic features did not add much value; it would be worthwhile to consider additional semantic features that might be useful for indoor/outdoor classification besides sky and grass. Diversifying the Bayesian network in such a way might help improve the contribution of computed semantic features. In particular, semantic features that are not easily detectable using the same low-level features could be considered. Future work also to be considered includes feature sharing. That is, using the same low-level feature set for semantic feature detection. For instance, the same color and texture features used for indoor/outdoor low-level feature classification could be used to train classifiers for grass and sky detection. In such a scenario, the low-level features are extracted once, then directed to different classifiers inferring different semantic content. This would represent a substantial reduction in computation time of the overall system.

8. REFERENCES

[1] A. Rosenfeld, "From Image Analysis to Computer Vision: An Annotated Bibliography, 1955-1979," *Computer Vision Image Understanding*, vol. 84, pp. 298-324, 2001.

[2] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-Based Image Retrieval at the End of the Early Years," *IEEE Trans. Pattern Anal. Machine Intell.*, pp. 1349-1380, December 2000.

[3] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and Effective Querying by Image Content," *J. Intell. Inform. Systems*, vol. 3, pp. 231-262, 1994.

[4] A. Pentland, R. W. Picard, and S. Scarloff, "Photobook: Tools for Content-Based Manipulation of Image Databases," *Proc. SPIE Storage Retrieval Image Video Databases II,* pp. 34-47, February 1994.

[5] J. R. Smith and S.-F. Chang, "VisualSEEk: A Fully Automated Content-Based Image Query System," *Proc. ACM Multimedia,* pp.87-98, Boston, MA, November 1996.

[6] S. Mehrotra, Y. Rui, M. Ortega-Binderberger, T. S. Huang, "Supporting Content-Based Queries Over Images in MARS," *Proc. IEEE Int. Conf. Multimedia Computing Systems,* pp. 632-633, Toronto, ON, June 1997.

[7] W. Y. Ma and B. S. Manjunath, "NeTra: A Toolbox for Navigating Large Image Databases," *Proc. IEEE Int. Conf. Image Process.,* pp. 568-571, Santa Barbara, CA, October 1997.

[8] R. W. Picard and T. P. Minka, "Vision Texture for Annotation," *Multimedia Systems*, vol.3, pp. 3-14, 1995.

[9] A. C. Loui and A. E. Savakis, "Automatic Image Event Segmentation and Quality Screening for Albuming Applications," *Proc. Int. Conf. Multimedia Expo*, New York, NY, 2000.

[10] Y. Gdalyahu, D. Weinshall, and M. Werman, "Self-Organization in Vision: Stochastic Clustering for Image Segmentation, Perceptual Grouping, and Image Database Organization," *IEEE Trans. Pattern Anal. Machine. Intell.*, vol. 23, pp. 1053-1073, October 2001.

[11] A. Vailaya, M. A. T. Figueiredo, A. K. Jain, and H. J. Zhang, "Image Classification for Content-Based Indexing," *IEEE Trans. Image Process.*, vol. 10, pp. 117-130, January 2001.

[12] M. Szummer and R. W. Picard, "Indoor-Outdoor Image Classification," *IEEE Int. Workshop Content Based Access Image Video Databases, ICCV '98*, 1998.

[13] S. Paek, C. L. Sable, V. Hatzivassiloglou, A. Jaimes, B. H. Schiffman, S.-F. Chang, and K. R. Mckeown, "Integration of Visual and Text Based Approaches for the Content Labeling and Classification of Photographs, *ACM SIGIR '99 Workshop Multimedia Indexing Retrieval*, Berkeley, Ca, 1999.

[14] A. Vailaya, M. Figueiredo, A. Jain, and H. J. Zhang, "Content-Based Hierarchical Classification of Vacation Images," *IEEE Multimedia Conf.*, Florence, Italy, 1999.

[15] A. Savakis and J. Luo, "Indoor vs. Outdoor Classification of Consumer Photographs," *Proc. IEEE Int. Conf. Image Process.*, Thessaloniki, Greece, September 2001.

[16] A. Guerin-Dugue and A. Oliva, "Classification of Scene Photographs from Local Orientation Features," *Pattern Recognit. Letters,* vol. 21, pp. 1135-1140, 2000.

[17] A. Vailaya, A. Jain, H. J. Zhang, "On Image Classification: City Images vs. Landscapes," *Pattern Recognit.*, vol. 31, pp. 1921-1936, December 1998.

[18] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image Indexing Using Color Correlograms," *Proc. IEEE Conf. Computer Vision Pattern Recognit.*, San Juan, Puerto Rico, pp. 762-768, 1997.

[19] R. W. G. Hunt, "Measuring Colour," Fountain Press, England, Third Edition, 1998.

[20] C.-H. Lee, B.-J. Moon, H.-Y. Lee, E.Y. Chung, and Y.H. Ha, "Estimation of Spectral Distribution of Scene Illumination from a Single Image," *J. Imaging Science Technology*, vol. 44, no. 4, pp. 308-320, 2000.

[21] Y.-I. Ohta, T. Kanade, and T. Sakai, "Color Information for Region Segmentation," *Computer Graphics Image Process.,* vol. 13, pp. 222-241, 1980.

[22] J. Luo, R. T. Gray, and H.-C. Lee, "Towards Physics-Based Segmentation of Photographic Images", *Proc. IEEE Int. Conf. Image Process.*, Santa Barbara, CA, 1997.

[23] M. Tuceryan and A. K. Jain, "Texture Analysis," *Handbook Pattern Recognit. Computer Vision,* C. H. Chen, L. F. Pau, and P. S. P. Wang, eds., ch. 2, pp. 235-276. Singapore: World Scientific, 1993.

[24] R. M. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. 67, pp. 786-804, May 1979.

[25] H. Weschler, "Texture analysis—A survey," *Signal Process.,* vol. 2, pp. 271-282, 1980.

[26] B. Julesz, E. N. Gilbert and J. D. Victor, "Visual discrimination of textures with identical third-order statistics," *Biol. Cybern.*, vol. 31, pp. 137-140, 1978.

[27] S. Marcelja, "Mathematical Description of the Responses of Simple Cortical Cells," *J. Opt.. Soc. Am.*, vol. 70, pp. 1297-1300, 1980.

[28] J. Beck, A. Sutter and R. Ivry, "Spatial Frequency Channels and Perceptual Grouping in Texture Segregation," *Computer Vision Graphics Image Process.*, vol. 37, pp. 299-325, November 1987.

[29] J. Mao and A. K. Jain, "Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models," *Pattern Recognit.*, vol. 25, pp. 173-188, May 1992.

[30] R. M. Rao and A. S. Bopardikar, "Wavelet Transforms: Introduction to Theory and Applications," Addison Wesley, 1998.

[31] S. G. Mallat, "A Theory of Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 674-693, July 1989.

[32] T. Chang and C.-C. J. Kuo, "Texture Analysis and Classification with Tree Structured Wavelet Transform," *IEEE Trans. Image Process.*, vol. 2, pp. 429-441, October 1993.

[33] M. Unser, "Texture Classification and Segmentation Using Wavelet Frames," *IEEE Trans. Image Process.*, vol. 4, pp. 1549-1560, November 1995.

[34] T. Randen and J. H. Husoy, "Filtering for Texture Classification: A Comparative Study," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 291-310, April 1999.

[35] R. R. Brooks, L. Grewe, and S. S. Iyengar, "Recognition in the Wavelet Domain: A Survey," *J. Electronic Imaging*, vol. 10, pp. 757-784, July 2000.

[36] S. G. Whittaker and J. B. Siegfried, "Origin of Wavelets in the Visual Evoked Potential," *Electroencephalogr. Clin. Neurophysiol.*, vol. 55, pp. 91-101, 1983.

[37] J. D. Villasenor, B. Belzer, and J. Liao, "Wavelet Filter Evaluation for Image Compression," *IEEE Trans. Image Process.*, vol. 4, pp. 1053-1060, August 1995.

[38] D. S. Taubman and M. W. Marcellin, "JPEG2000: Image Compression Fundamentals, Standards and Practice," Kluwer Academic Publishers, 2002.

[39] A Laine and J. Fan, "Texture classification by wavelet packet signatures," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1186-1191, November 1993.

[40] E. Salari and Z. Ling, "Texture segmentation using hierarchical wavelet decomposition," *Pattern Recognit.*, vol. 28, pp. 1819-1824, December 1995.

[41] G. Wu, Y. Zhang, and X. Lin, "Wavelet Transform-based Texture Classification with Feature Weighting," *Proc. IEEE Int. Conf. Image Process.*, Kobe, Japan, pp. 435-439, 1999.

[42] A. Mojsilovic, M. V. Popovic, and D. M. Rackov, "On the Selection of an Optimal Wavelet Basis for Texture Characterization," *IEEE Trans. Image Process.*, vol. 9, pp. 2043-2050, December 2000.

[43] H.-C. Lee and D. R. Cok, "Detecting Boundaries in a Vector Field," *IEEE Trans. Signal Process.*, vol. 39, pp. 1181-1194, May 1991.

[44] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 679-698, November 1986.

[45] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining Knowledge Discovery*, vol. 2, pp. 1-43, 1998.

[46] V. Cherkassky and F. Mulier, "Learning from Data," John Wiley & Sons, 1998.

[47] C. Cortes and V. Vapnik, "Support Vector Networks," *Machine Learning,* vol. 20, pp. 273-297, 1995.

[48] T. Joachims, "Text Categorization with Support Vector Machines," Tech. Rep. LS VIII No. 23, University of Dortmund, 1997.

[49] E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines: An Application to Face Detection," *Proc. IEEE Computer Vision Pattern Recognit.,* pp. 130-136, 1997.

[50] http://www.ai.mit.edu/projects/cbcl/software-datasets

[51] A. Vailaya and A. Jain, "Detecting Sky and Vegetation in Outdoor Images," *Proc. IS&T/SPIE Symp. Electronic Imaging Science Technology,* 2000.

[52] J. Luo and S. P. Etz, "A Physical Model-Based Approach to Detecting Sky in Photographic Images," *IEEE Trans. Image Process.,* vol. 11, pp. 201-212, March 2002.

[53] N. Sprague and J. Luo, "Clothed People Detection in Still Images," *Proc. Int. Conf. Pattern Recognit.,* 2002. (To appear)

[54] J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference," Morgan Kaufmann, 1988.

[55] A. Singhal and C. Brown, "Dynamic Bayes Net Approach to Multi-Modal Sensor Fusion," *Proc. SPIE Conf. Sensor Fusion Decentralized Control*, vol. 3209, Pittsburgh, PA, 1997.

[56] J. Luo, A. Singhal, S. Etz, and R. Gray, "Performance Scalable Computational Approach to Main Subject Detection in Photographs," *Proc. SPIE Conf. Human Vision Electronic Imaging*, vol. 4299, San Jose, CA, 2001.

[57] J. Luo, A. Savakis, A. Singhal, and S. Etz, "On the Application of Bayesian Networks to Semantic Understanding of Consumer Photographs, " *Proc. IEEE Int. Conf. Image Process.*, Vancouver, Canada, 2000.

APPENDIX

```
function F = lst(f)
% Convert 8-bit RGB image to LST color space.
%
% Usage:    F = lst(f)
%
% Author:   Navid Serrano


f = double(f);
k = 255/max(max(max(f)));

F(:,:,1) = (k/sqrt(3)) * (f(:,:,1) + f(:,:,2) + f(:,:,3));
F(:,:,2) = (k/sqrt(2)) * (f(:,:,1) - f(:,:,3));
F(:,:,3) = (k/sqrt(6)) * (f(:,:,1) - 2*f(:,:,2) + f(:,:,3));


function h = imghist(im,N)
% Compute the pixel value histogram of an image using N bins.
%
% Usage:    h = imghist(im,N)
%
%           im - source image
%           N  - histogram bins
%
% Author:   Navid Serrano


im = double(im);
h = hist(im,N);
h = sum(h');



function [e,m,a] = canny(f,T1,T2)
% Compute the edge magnitude and direction of an image.
%
% Usage:    [e,m,a] = canny(f,T1,T2)
%
%           f - source image
%           T1 - low threshold
%           T2 - high threshold
%
% Author:   Navid Serrano


warning off

[rw,cl,ch] = size(f);
f = double(f);


% Define 3x3 Gaussian smoothing kernel
g = [0.0001 0.0070 0.0001;0.0070 0.9718 0.0070;0.0001 0.0070 0.0001];


% Define 3x3 Prewitt derivative filters
hx = [-1 -1 -1;0 0 0;1 1 1];
```

```
hy = [-1 0 1;-1 0 1;-1 0 1];


% Smooth image using Gaussian filter and compute partial derivatives
for k=1:ch

   s(:,:,k) = filter2(g,f(:,:,k));
   dx(:,:,k) = filter2(hx,s(:,:,k));
   dy(:,:,k) = filter2(hy,s(:,:,k));

end


% Compute edge magnitude and direction using PCA
for i=1:rw
   for j=1:cl

      if ch > 1
         A = [dx(i,j,1) dy(i,j,1);dx(i,j,2) dy(i,j,2);dx(i,j,3)
dy(i,j,3)];
      else
         A = [dx(i,j) dy(i,j,1)];
      end

      [V,D] = eig(A'*A);

      D = D(find(D));
      m(i,j) = max(D);
      index = find(D==m(i,j));
      a(i,j) = atan(V(2,index)/V(1,index));

   end
end


% Quantize angles to four zones
q = floor(4*(a+pi/2)/pi) + 1;


% Define 8-neighbor general coordinates
di = [ 1 0 -1 -1 -1  0  1 1];
dj = [ 1 1  1  0 -1 -1 -1 0];


% Find candidate boundary points and eliminate weak edges
e = zeros(rw,cl);

for i=2:rw-1
   for j=2:cl-1

      if m(i,j)>m(i+di(q(i,j)),j+dj(q(i,j))) & m(i,j)>m(i-di(q(i,j)),j-
dj(q(i,j)))
         if m(i,j) > T1
            for k=1:8
               if m(i+dj(k),j+dj(k)) > T2
                  e(i,j) = 1;
                  break;
               end
            end
         end
      end
```

```
   end
end

warning on



function h = edgehist(e,a,bins)
% Construct an edge direction histogram.
%
% Usage:    h = edgehist(e,a,bins)
%
%           e - edge map (from Canny operator)
%           a - edge angle
%           bins - number of bins in histogram
%
% Author:   Navid Serrano


% Extract image dimensions
[r,c] = size(e);


% Compute total number of points in image
Np = r * c;


% Eliminate points not considered edges
a = a .* (e>0);
a = find(reshape(a,1,Np));


% Compute number of edge points in image
Ne = length(a);


% Construct edge direction histograms
if Ne > 0
   h = hist(a,bins) / Ne;
   h(bins+1) = (Np - Ne) / Np;
else
   h = [zeros(1,bins) 1];
end



function Wc = wavcoef(f,N,wname)
% Extract wavelet decomposition coefficients (requires Matlab wavelet
toolbox).
%
% Usage:    [Wc] = wavcoef(f,N,wname)
%
%           f - source image
%           N - decomposition levels
%           wname - wavelet filter name (use Matlab denominations)
%
% Author:   Navid Serrano


[r,c] = size(f)
```

```
[C,S] = wavedec2(f,N,wname);
Wc{1} = appcoef2(C,S,wname,N);
j = 1;

for i=N:-1:1

   Wc{3*i+1} = detcoef2('d',C,S,j);
   Wc{3*i} = detcoef2('v',C,S,j);
   Wc{3*i-1} = detcoef2('h',C,S,j);

   r = r / 2;
   [r2,c2] = size(Wc{3*i});
   e = floor((r2-r)/2);

   if odd(r2)
      Wc{3*i+1} = Wc{3*i+1}(e+2:r2-e,e+2:c2-e);
      Wc{3*i} = Wc{3*i}(e+2:r2-e,e+2:c2-e);
      Wc{3*i-1} = Wc{3*i-1}(e+2:r2-e,e+2:c2-e);
   else
      Wc{3*i+1} = Wc{3*i+1}(e+1:r2-e,e+1:c2-e);
      Wc{3*i} = Wc{3*i}(e+1:r2-e,e+1:c2-e);
      Wc{3*i-1} = Wc{3*i-1}(e+1:r2-e,e+1:c2-e);
   end

   j = j + 1;

end

if odd(r2)
   Wc{1} = Wc{1}(e+2:r2-e,e+2:c2-e);
else
   Wc{1} = Wc{1}(e+1:r2-e,e+1:c2-e);
end



function [c,t,e] = iocfeatures(im,class,blk,T1,T2)
% Extract color, texture, and edge direction features.
%
% Usage:    [c,t,e] = iocfeatures(im,blk,T1,T2)
%
%           im - source image
%           blk - tessellation (e.g. for 4x4 tessellation, blk=4)
%           T1 - low threshold for Canny edge selection
%           T2 - high threshold for Canny edge selection
%
% Author:   Navid Serrano


% Extract image dimensions
[rw,cl,ch] = size(im);


% Convert RGB image to LST space
I = lst(im);


% Compute edge magnitude and direction
[E,M,A] = canny(im,T1,T2);
```

```
% Extract color and edge features for desired tessellation (or full
image)
brw = rw/blk;
bcl = cl/tblk
b = 1;

for i=1:brw:rw
   for j=1:bcl:cl

      e(b,:) = edgehist(E(i:i+brw-1,j:j+bcl-1),A(i:i+brw-1,j:j+bcl-
1),36);
      c(b,:) = [imhist(I(i:i+brw-1,j:j+bcl-1,1),16)
                imhist(I(i:i+brw-1,j:j+bcl-1,2),16)
                imhist(I(i:i+brw-1,j:j+bcl-1,3),16)]';
      b = b + 1;

   end
end


% Compute 2-level DWT using biorthogonal 3/5 filter
W = wavcoef(I(:,:,1),2,'bior2.2');


% Filter LL1 coefficients using Laplacian
W{1} = filter2([-1 -1 -1;-1 8 -1;-1 -1 -1]/9,W{1});


% Extract wavelet texture features for desired tessellation
for k=1:7

   [rw,cl] = size(W{k});
   brw = rw/blk;
   bcl = cl/blk;
   b = 1;

   for i=1:brw:rw
      for j=1:bcl:cl

         t(b,k) = sum(sum(abs(W{k}(i:i+brw-1,j:j+bcl-1)).^2));
         b = b + 1;

      end
   end

end


% Append indoor/outdoor class to each subblock (or full image)
c(1:blk^2,49) = class;
t(1:blk^2,8) = class;
e(1:blk^2,38) = class;
```